Investigation of the use of meshfree methods for haptic thermal management of design and simulation of MEMS

Jean-Baptiste Munier

Submitted for the degree of Doctor of Philosophy

Heriot-Watt University
School of Engineering and Physical Sciences
December 2016

Abstract

This thesis presents a novel approach of using haptic sensing technology combined with virtual environment (VE) for the thermal management of Micro-Electro-Mechanical-Systems (MEMS) design. The goal is to reduce the development cycle by avoiding the costly iterative prototyping procedure. In this regard, we use haptic feedback with virtual prototyping along with an immersing environment. We also aim to improve the productivity and capability of the designer to better grasp the phenomena operating at the micro-scale level, as well as to augment computational steering through haptic channels.

To validate the concept of haptic thermal management, we have implemented a demonstrator with a user friendly interface which allows to intuitively "feel" the temperature field through our concept of haptic texturing. The temperature field in a simple MEMS component is modeled using finite element methods (FEM) or finite difference method (FDM) and the user is able to feel thermal expansion using a combination of different haptic feedback.

In haptic application, the force rendering loop needs to be updated at a frequency of 1Khz in order to maintain continuity in the user perception. When using FEM or FDM for our three-dimensional model, the computational cost increases rapidly as the mesh size is reduced to ensure accuracy. Hence, it constrains the complexity of the physical model to approximate temperature or stress field solution. It would also be difficult to generate or refine the mesh in real time for CAD process.

In order to circumvent the limitations due to the use of conventional mesh-based techniques and to avoid the bothersome task of generating and refining the mesh, we investigate the potential of meshfree methods in the context of our haptic application.

We review and compare the different meshfree formulations against FEM mesh based technique. We have implemented the different methods for benchmarking thermal conduction and elastic problems. The main work of this thesis is to determine the relevance of the meshfree option in terms of flexibility of design and computational charge for haptic physical model.

To my parents and Charmaine.

Acknowledgements

This research was sponsored by the Scottish manufacturing institute (SMI) and the EPSRC-funded Grand Challenge Project " 3D MINTEGRATION". I would like to thank Professor Marc Desmulliez for giving to me the opportunity to do research within the MISEC group. I also thank Professor Omar Laghrouche for his introduction to the realm of FEM and his friendship. I thank my parents for their understanding and support along all those years. Finally I would like to dedicate this thesis to Charmaine for all her support and love.

ACADEMIC REGISTRY
Research Thesis Submission

Name*:*	Jean-Baptiste Munier		
School:	School of Engineering and Physical Sciences		
Version: *(i.e. First, Resubmission, Final)*	Final	Degree Sought:	Doctor of Philosophy (PhD)

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6) I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:*		Date:	12/01/2017

Submission

Submitted By *(name in capitals):*	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals):*			
Method of Submission *(Handed in to SSC; posted through internal/external mail):*			
*E-thesis Submitted (**mandatory for final theses)***			
Signature:		Date:	

Please note this form should be bound into the submitted thesis.

Contents

List of Tables

List of Figures

Chapter 1 Introduction

The global market for MEMS devices and production equipment was worth $12.8 billion in 2015. This market is growing fast and is expected to reach $21.9 billion by 2020. Over the last 3 decades, many fabrication processes and techniques for building Microsystems, such as dry anisotropic etching for bulk and surface micro-machining, double-sided lithography and PDMS soft lithography, have been developed. From accelerometers for airbag systems, pressure sensors, humidity sensors to microfluidics for embedded diagnosis system and DNA detection platforms; the range of applications continues to grow, constantly driving the need for new technology and design solutions.

Although macroscopic and microscopic worlds are governed by the same physical laws, the relative importance of individual physical effects is greatly affected by scaling. Designing Microsystems is more than just miniaturising macroscopic systems. On top of the scaling effects such as the predominance of magnetic force over relative weight due the mass, micro-scale effects such as Brownian noise, surface tension or Casimir effect have to be considered from the design process.

Microsystems often involve more than one energy domain. Mechanics and electrical effects are usually the cornerstone of many sensors and actuators but many other effects such as thermal, fluidic, optics and even biology can be involved. An understanding of the interactions between the effects of the different domains operating on the whole microsystems is key in the design process. Due to the complexity of the analysis of such problems, designers are often looking for simpler models or macromodels [256], that preserve the relationship between the macroscopic parameters and allow for optimisation and prediction of failures while speeding up the design process.

Nowadays, component and system design relies on the use of Computer-Aided Design (CAD) tools. It is estimated that the market for CAE is in the $1-$2 billion range and CAD is in the $5-$10 billion range [126]. Dedicated software for MEMS design such as ConventorWare [69], IntelliSuite, SESES [94], CFD-ACE+ [79], MEMS Xplorer [266], are able to handle multiple energy domains in the analysis.

The use of CAD tools has helped to reduce the cost and time for product development along with speeding up device optimisation right from the prototyping stage. The simulation of fabrication processes and device behaviour as a logical loops (see figure 1.1), enables us to prepare, run, optimise and analyse experiments, which helps the development of optimal manufacturing recipes and device targets.

Numerical methods such as Finite Element Method (FEM) and Boundary Element Method (BEM) are key in the performance of CAD tools. They rely on mesh to carry out the analysis process. In the design flow, engineers model their design using the CAD interface, whereby a preprocessor will automatically generate the mesh required to carry out the analysis stage. Before the analysis can be done, what is solved is actually a complex discretization problem through the meshing of the CAD model. This meshing requires insight from engineers to decide on a sensible accuracy during the analysis stage. If the analysis finds faults, the process will go back to the modelling

at the CAD environment stage. This iterative design process is time consuming and has to be repeated if the prototype of the final product fails at the test phase.

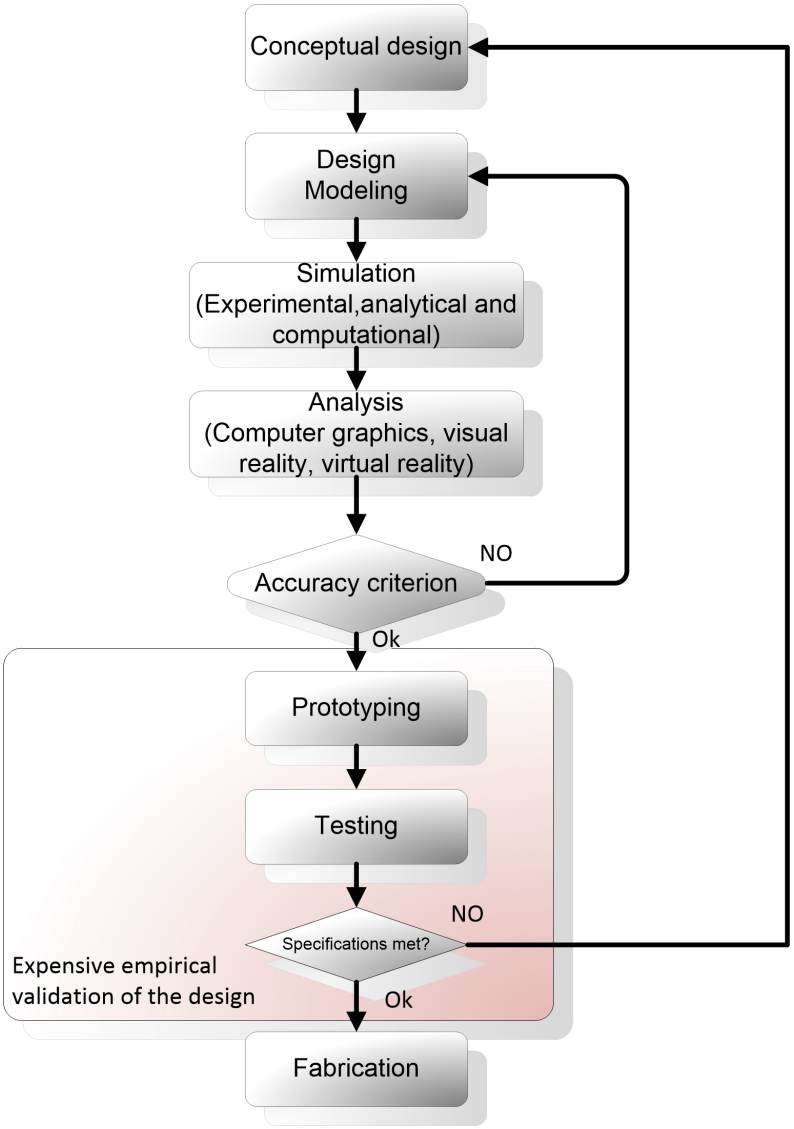


Figure 1.1: Product development cycle

Due to the rising cost of well-trained engineers and the decreasing cost of computation resources and CPU time, companies are increasingly concerned about the man-hours engineers spend on a project. The industry is seeking for solutions to reduce the whole design iterative process and the cost associated with repetitive prototyping (figure 1.1). In that regard, haptic-based design combined with virtual prototyping has garnered interest because it has the potential to decrease design time, which will help reduce the engineering process costs [53, 84].

1.1 Motivation

The aim of this project is to use haptic sensing technology and virtual prototyping as a new optimised way of designing MEMS. We look to haptic technology for increasing computational steering which is a valuable mechanism for scientific investigation in which the parameters of a running computer program can be altered and the results visualized immediately [286, 297]. The combination of haptic sensing technology and virtual environment immersion has been proven to

provide tangible costs benefits, in terms of reduced manufacturing time and cycles [84]. By fully immersing the designer in a virtual MEMS world, the goal is to bring back to the macro-world, microscopic forces that are hard to assess. Predominance of magnetic field or thermal effects over mass of object can be integrated through haptic feedback channels to assist naturally designer choices.

This thesis deals with the prospecting of solutions for implementing a haptic tool for the thermal management of MEMS and their packaging at the design stage. Thermal effects such as Peltier, are commonly used in MEMS sensors and actuators. Heat also has a direct impact on the overall performance, reliability and efficiency of MEMS devices. Thermal stress is a recurrent problem encountered in fabrication and packaging of MEMS. It can lead to several failures like cracks, fracture and delamination problems, which increases the complexity of the analysis and simulation of the design.

In haptic applications, heavy computation is needed for graphical display and to render interactions such as collision detection and the forces feedback. Hence it limits the resources for computation of the model's physics such as temperature or deformation.

In many practical haptic applications, the physical behaviour is built over a geometric model for simplicity. The response of the system can also be pre-characterised and stored in look-up tables to ensure real-time computation of the forces and effects to render through the user haptic interface. These techniques do not apply well for our use of haptic technology and virtual model for design and simulation of MEMS. Contrary to pure graphical applications with geometric only deformations, our project requires the accurate simulation of the model physics.

Physics based models that use numerical methods such as FEM are preferred for that matter. To achieve good accuracy, a varying density of elements is used to model complex geometry problems or localised strong gradient of the field function. This adaptivity feature in the design and simulation process is essential but requires re-meshing or refinement of the mesh as the model shape is modified by the designer. For three-dimensional applications, the computational cost grows rapidly as the mesh size is reduced, making it difficult to generate an optimum mesh during the real time CAE process. Thus the demand for extreme computational efficiency for haptic applications limits the complexity of the computational models used for MEMS simulation.

In this regard, this thesis focuses on investigating the potential ability of meshfree methods to circumvent the limitations of mesh-based method for our haptic application. The absence of the requirement for mesh appears as a very attractive feature. For haptic design, building approximation of field solutions using only the nodes would provide an easy solution to the problem linked to re-meshing as well as facilitate the modification of the geometry of the system without the need for the mesh update and refinement process that is computationally expensive.

1.2 Contribution

The contributions of the thesis are the following:

- Demonstrate the benefits of haptic sensing technology for MEMS design and simulation.
- Review the key features for the successful development and implementation of a haptic thermal management design tools for MEMS.

- Review and find suitable methods for modelling MEMS components as well as environmental impact on fabrication and packaging.
- Implement a user friendly demonstrator for haptic thermal prospecting of a simple component.
- Highlight the problematic of the limitations of mesh-based methods in the simulation of the thermal and mechanical physics for the MEMS component in real-time.
- Propose and assess haptic texturing as a rendering method for thermal feedback.
- Investigate the potential of meshfree mehods for modelling MEMS for implementation within the haptic environment. Present a well-rounded analysis of the advantages and drawbacks of the meshfree methods through the application of series of test problems.
- Produce a series of meshfree codes for basic problems related to the thermal management of MEMS design. This involves elastic problems for beam problems, linear and non-linear thermal conduction problems in 2D and 3D. These codes are implemented in a Matlab and C++ format for future applications and implementation in the OPENHAPTIC framework.

1.3 Thesis outline

The layout and purpose of this thesis is presented schematically in figure 1.2 p 6. The outline of the thesis is as follows:
Chapter 2 presents the haptic technology and the potential benefits for MEMS design. A review of haptic technology is presented as well as the necessary conditions to implement it. We analyse several key aspects particularly important for haptic implementation. We present how human physiology and visual perception accuracy condition the VE's complexity for an optimal haptic thermal rendering. The different types of haptic interactions with 3D objects in VE are described along with collision detection methods. Lastly the different methods for haptic rendering and related issues are presented. Characteristic features of the different methods as well as potential issues are addressed.
Chapter 3 presents the implementation of a haptic thermal management demonstrator. We review the different methods for modelling MEMS components in VE. Haptic texturing is presented and tested as a solution for the thermal rendering. The tests using mesh based FDM and FEM for the physical model highlights the difficulty in using mesh based model for our application.
Chapter 4 is an introduction and overview of meshfree methods. We present the different techniques for building shape functions over a scattered set of data which is the cornerstone of meshfree formulation. From the Smooth Particle Hydrodynamic (SPH) method, the Reproducing Kernel Method (RPKM) and Moving Least Square (MLS) method are derived and introduced. Radial Basis functions (RBFs) along with the Radial Point Interpolation Method (RPIM) are also detailed. We study MLS and RPIM shape functions for scatter data fitting problems, to assess the range of optimal parameter values that should be used for benchmark tests.
Chapter 5 details the implementation procedure of the meshfree methods based on Galerkin weak formulation such as the Element Free Galerkin Method (EFGM). EFGM, RPIM as well as the local formulation Meshless Local Petrov Galerkin Method (MLPG) are introduced. Implementations

for a microswitch simplified as a cantilever model is carried out in Matlab code and C++. Comparison with regular FEM is done for assessing the general performance of the methods. The chapter also addresses the issue in the implementation of the method such as multi material or geometry discontinuities and the integration issue for weak formulated based methods.

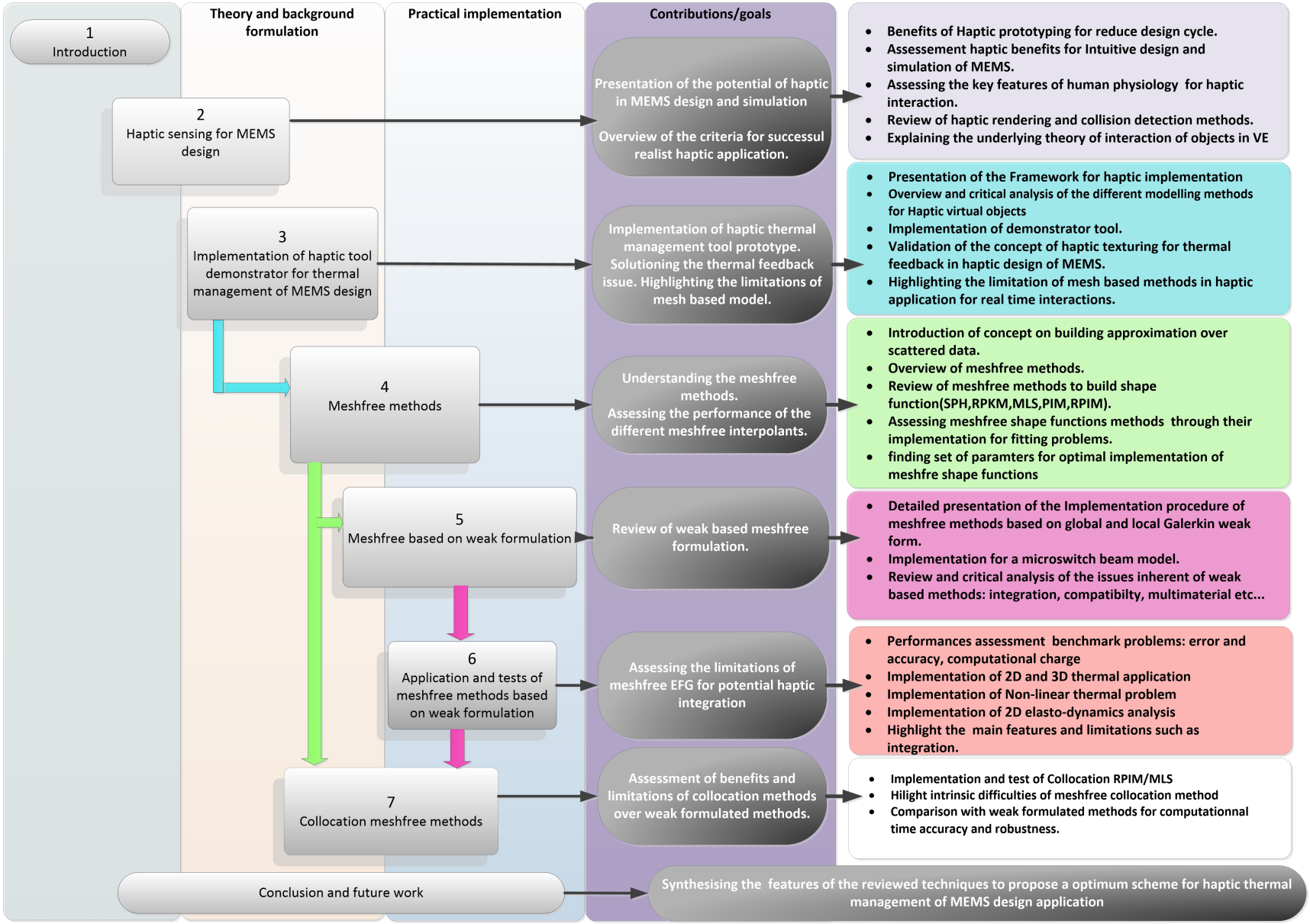
Chapter 6 focuses on the implementation of meshfree weak based methods for different thermal and mechanical benchmark problems. Linear and non-linear heat conduction for 2D and 3D problems are covered. Thermal effects are widely used in actuators and sensors. They are also of primary importance as many of the MEMS fabrication involve thermal processes such as baking, cooling, curing and exposition to radiation.

Beams are another very common structure met in MEMS sensors. Their study is a good benchmark choice as both static, dynamic vibration characterisation are involved in designing MEMS structures. Comparison with our own standard FEM code is carried out.

Chapter 7 introduces the methods based on collocation formulation as a good way to alleviate the cost of integration of weak based methods. The methods are implemented in Matlab code and C++ for several problems in order to compare their performances with the previous methods. Pros and cons of the methods are highlighted.

Chapter 8 presents the conclusion on the potential of meshfree methods for haptic thermal management of MEMS design application. Based on the work of this thesis, we present ideas for future developments and new direction of research that could solve the issues and limitations encountered through the work of this thesis.

Figure 1.2: Thesis layout



Chapter 2 Haptic sensing for MEMS design

"Haptic" relates to the action of sensing and manipulating objects through touch [267]. The aim of haptic technologies is to immerse the user in a virtual reality environment (VRE) by displaying and rendering forces through a dedicated device in response to the user interaction. It involves tactile cues such as surface textures and vibrations. Haptic also includes kinesthesia (proprioception), the ability to perceive one's body position, movement, shape, inertia and weight [111]. VRE or virtual environment (VE) is a interactive computer-generated 3D synthesised environment in which the user is immersed. Haptic devices are used as interfaces and add new ways of tactile interaction with virtual reality simulations. The general requirements for haptic interaction can be summarised as follows:

- The VE must be constructed naturally.
- The user should manipulate virtual objects freely, and in real-time.
- The user should feel immersion. Concept of "enveloping environment" and degree of involvement and integration of the user into the VE, is often referred to as the degree of presence (DOP) [279].

The DOP is a concept describing more than immersion but "a mental state in which a user feels physically present within the VE". In a haptic interface, the user moves the haptic device, which will react by generating a force feedback if an object is encountered in the VE. This process of generating an appropriate force feedback is called haptic rendering. The research in haptic modelling and simulation focuses on the development off haptic interfaces for the interaction of the designer with the VE and numerical methods to render the physics within the VE.

2.1 Haptic for MEMS design and simulation

The development of CAD has followed the rise of computer technology to become a key step in any modern engineering design. In micro-engineering, CAD is understood as the exploration, by means of simulation, of fabrication processes and micro-device behaviour. The development of computer programs called simulators, starts with the understanding of physical phenomena and their modelling, mapped onto mathematical representations which can be handled by calculating machines. CAD tools (see [296] for an exhaustive list) help to improve existing technologies and simultaneously reduce research and development time, which has a direct impact on cost reduction. Modern CAD tools aims to

- Increase the quality of products or processes,
- Accelerate the design cycle while lowering the cost of development,
- Enhance the control over fabrication process,

- Predict and control the life-cycle of the product by incorporating robust error estimate and also risk assessment control point.

Nowadays CAD environment possess tools for each step of the product life cycle management, from conception to manufacture as shown in figure 2.1.

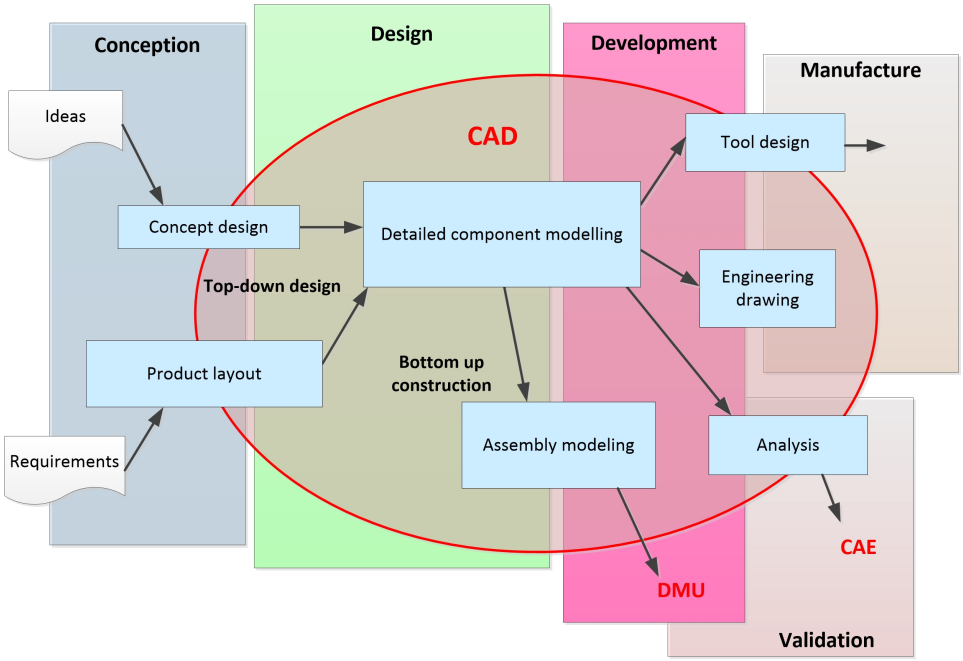


Figure 2.1: Computer aided design

MEMS technology is still based on traditional design tools borrowed from microelectronics design and the traditional product development cycle design-build-test approach. The fabrication process still relies heavily on the trial and error approach (figure 1.1 p.2). In this process a draft design is produced through CAD then a prototype for manufacturing and testing is created. In the case that specifications are not met, the design has to be revised and a new prototype will be generated again then tested. This iteration procedure is repeated until the specifications are met. This approach is time consuming due to the different micro-engineering processes involved. Prototypes in MEMS become costly through repetitive iteration as many resources for MEMS building process such as material like silver, copper, Si or wafer are expensive.
For this reason a simulation driven product development cycle (figure 2.2) to avoid iteration of prototyping is highly sought for by the industry [53]. The challenge is to merge the whole prototyping and testing within the VE.
The idea is to use a virtual platform in order to fully digitises the entire process from conception and production to maintenance and recycling, using a common, constantly-updated digital mock-up. This guarantees reduced development cycles and tooling/prototyping costs, improves quality and virtually eliminates assembly and rework issues.
The combination of haptic sensing technology and virtual environment immersion (figure1.1)has been proven to provide tangible costs benefits, in terms of reduced manufacturing time and cycles [84]. Figure 2.3 illustrates the benefits of using virtual prototyping for the design cycle of footwear.

Other successful application of this approach has been made by Dassault Systèmes with their Product Lifecycle Management (PLM) solution package for the development of the Falcon 7X

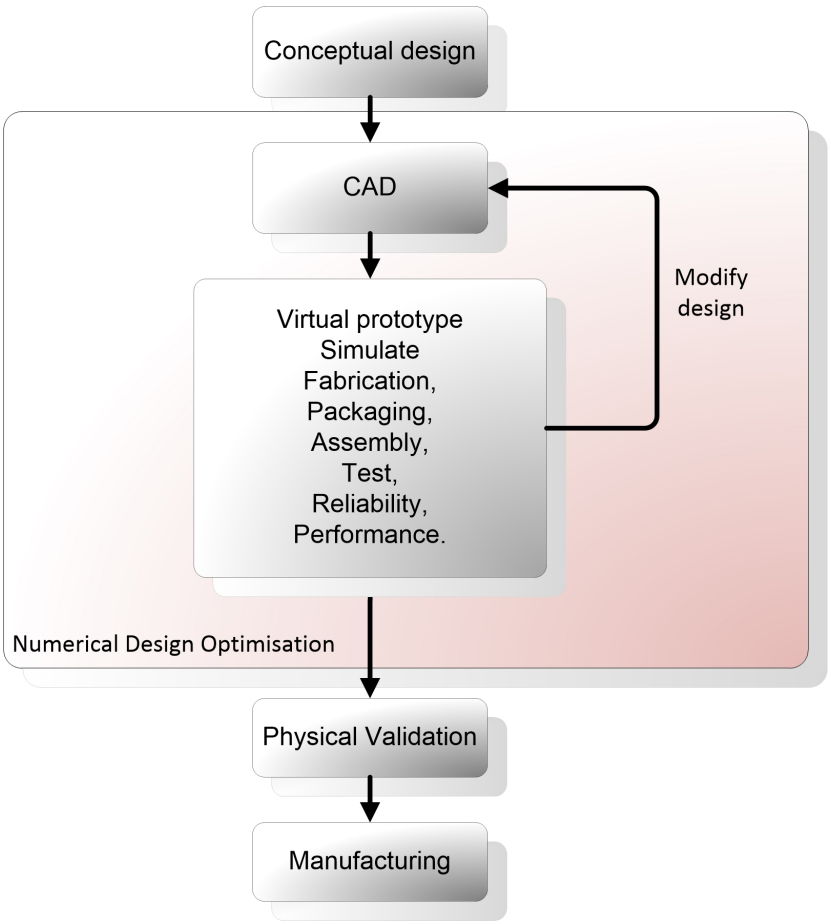


Figure 2.2: Simulation driven product development cycle

[71]. They introduced a virtual development platform to the aviation industry allowing partners and suppliers from around the world to design a new product, concurrently and in real time without physically leaving their workplace. The system is based on advanced 3D tools such as CATIA V5/V6 and Enovia PLM. Their unified data management approach made it possible to apply "zero mock-up" and "zero prototype" concepts. The first plane built was directly delivered to their customer after a flight test.

Haptic research is an evolution in digital tools that combines physical and digital modelling. The main advantage of haptic technologies is the user's ability to fully interact wither virtual prototype through the sense of touch. The main idea in using haptic technology in design is to rely more on creative idea than pure mathematical approach by directly modelling this idea through haptic interface into a 3D virtual prototype. Obviously this can only be achieved by well designed interface that match and transfer information efficiently and reliably with high bandwidth between the human sensory and the machine data. Another requirement is a powerful solution for the virtual model in order to assists the designer in the materialisation of his idea. Thus the detailed simulation of the physics behaviour of component is necessary.

Haptic technology increases the information flow i.e. bandwidth between the computer and the user which can "touch" and manipulate object in the VE in real time. This type of human computer interaction (HCI) is much similar to our natural way of interaction with objects in our real world. 3D manipulation techniques have a great impact on designer's efficiency, due to the better understanding of manipulation techniques and relevant design issues. Another benefit of this more natural way of interaction is the sense of comfort which convey more productivity [53]. In

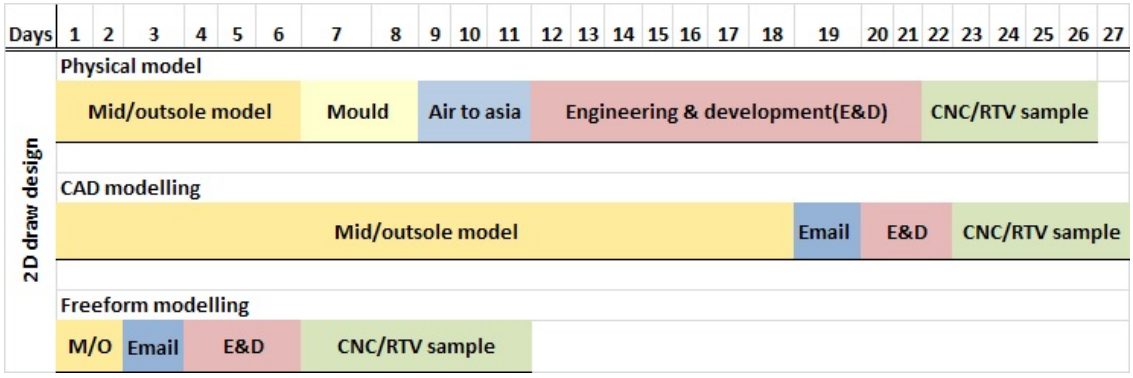


Figure 2.3: Benefit of virtual prototyping with haptic design free form for product development cycle time

MEMS design, haptic sensing brings back to the macro-world microscopic forces that are hard to assess by the designer or the predominance of magnetic field or thermal effects over mass of object. The haptic technology can also increase the data feedback due to the augmented channels. Effects such as stiction and vibration can be used for this purpose. The haptic environment should not be seen as a standalone module but truly embedded in the CAD and CAE framework in order to offer a new level of productive collaborative tool. Haptic devices are also useful for tasks where visual information is not sufficient and may induce unacceptable manipulation errors, for example during assembly process. An overview of the advantages of using haptic technology in VE is shown in table 2.1 p.11. The idea is to have the design and prototyping processes of MEMS be more straightforward and instinctive by using surface effects rendering (stiffness, hardness, friction, stiction) and force-momentum rendering. Thus, designers can experience these effects in real-time through the simulation process and interact directly with their prototype.

2.2 Physiology criteria for efficient haptic interface

Haptic technology is based on the use of the physiology of the sense of touch. Displaying and rendering force feedback for user interaction necessitates a good understanding of the human sensory and motor physiology. The cortex sensory area in the brain processes information from mechanoreceptors from the fingers tip [239]. The result is the feel of touch, pressure stretching and motion [138]. Understanding this chain is key in optimizing the human interaction and reliability while minimizing the risk for the user. The VE should be designed with the aim to immerse the user into the VRE without perception of system's limitations. The transparency defining how well the user actions are transformed into the VE, and the realism of the VE are critical in enhancing human performance efficiency. To achieve this, the haptic interface and rendering algorithm must address the limitations of human sensory systems. The features which govern our haptic application is the human fingertip physiology and the mechanism of human visual perception.

2.2.1 Human physiology for contact and force perception

Humans perceive force simultaneously through

- The sense of touch such as the perception of the various skin stimuli e.g. roughness, vibration, temperature

Application	Needs	Benefits of Haptic
CAD	Assistance and guidance for assembly process.	Manufacturing of mould at micro-scales for LIGA processes.
	Conception.	Sense surface, shape of components, deformation.
	Tolerance.	Perception of dominant effect at micro and meso-scales.
Virtual prototyping	Replacement of physical prototype.	Perception of the physics of the digital modelling.
	Enhance productivity.	Designers assistance through movement and force feedback. Improvement of manufacturing time. Increase data information flow between user and computer.
Visualization	Analyses of any parts of the system.	Improve understanding of the local comportment of model.
	Ergonomic environment.	Scale-up or down of the designer POV by immersive environment.
Maintenance	Verification. Diagnosis.	Security (VE). Unified Data system. Computational steering for fastest analysis of default.
Training	Simulation procedure scenario to improve understanding.	Force-feedback. Sense of gravity inertia.
	Application for manipulation.	immersive for critical environment e.g. non invasive surgery [276], wire bonding. Motion of components.

Table 2.1: Benefit of haptic technologies for PLM

- Kinesthesia deals with movement and force perceived through the muscle structure. Exteroception is the sensitiveness to stimuli outside the body such as mass, roughness of foreign object. Proprioception deal with self position awareness. The sensory system within muscles conveys positional information to the brain.

First the touch receptors provide information about the surface geometry and texture. Then, as the application force increases, kinesthesia will join in the loop by treating the information relative to the position, motion of the different limbs and the magnitude of forces applied. In our application, we are primarily interested in the sensory receptors channel located in the human hand such as the mechanoreceptors, thermo-receptors, nocioreceptors and proprioreceptors. The mechanoreceptors are involved when there is an interaction between objects and the skin. They are responsible for the treatment of the spatial, temporal and strength information of the force applied. We are interested in tactile feedback such as vibration, stiffness, roughness, fluttering and stroking. The receptors involved are the Meissner, Merkel disks and Pacinian corpuscles. For our haptic interface, the key perceptual cues are:

- Force sensing under quasi-static and dynamic conditions.
- Pressure perception.

- Position sensing resolution.
- The level of stiffness required for rigidity simulation.

In term of performances the main issues are:

- The maximum force humans can produce through the haptic chain.
- The precision with which humans can control a force.
- The control bandwidth of force.
- The ergonomic and comfort.

For the user to perceive the forces displayed by the device smoothly and to provide a meaningful tactile-feedback, the force display resolution of the device should match or exceed the human sensing resolution. Effective scaling system must provide high transparency to convey the most of what can be sensed by humans and minimise noise. It must have a large dynamic range since human motor and contact sensors deal with 4 to 5 orders of magnitude. They must have wide bandwidth since the haptic capabilities cover from Dc to about 1 kHz. The dynamic range is related to the force resolution of the interface. It quantifies the ratio of the largest to the smallest forces that can be commanded, and thus sensed, through the interface. Bandwidth refers to the capacity of the interface to convey mechanical signals over a large frequency range, in order to render complex phenomena, such as Brownian motion.
Overall, the fingertip information capacity is 10^2 bts/sec with 5ms of temporal acuity. Compared to the eye which has up to a 10^9 bits/sec capacity, the fingertip has a smaller capacity. However the temporal acuity of the eyes is only 25 ms resulting in more latency in signal treatment than the one for the touch.
For the successful implementation of our haptic solution for design and optimizing the rendering cue, there needs to be proper characterization of the bandwidth, threshold, period and intensity of the signal that has to be perceived by the mechanoreceptors of the hand. In the following section, description of the bandwidth and threshold for those chains of receptors are studied.

2.2.1.1 Bandwidths of sensory receptors

Human receptor bandwidth defines the resolution and sensitivity of the sensors and their spatial features of processing. For meaningful fingertips perception the minimum kinesthetic bandwidth is estimated to be in the range 20-30 Hz [263].
The tactile bandwidth is much larger with a range of 10Hz to 10KHz. This is equivalent to the skin vibration bandwidth for meaningful manipulative tasks. The human control bandwidth, 5 to 10 Hz, is similar to the time for the human to move his limbs or fingers comfortably in response to stimulus. However the response is correlated to state of comfort or type of stimuli. For uncomfortable signals, the control bandwidth will be in the range of 1 to 2 Hz whereas for repetitive patterns and tasks(periodic signal), the bandwidth will be from 2 to 5 Hz. In the case of reflex it can be up to 10Hz.
Mechanoreceptors such as the Merkel discs possess a larger bandwidth of 5 to 15 Hz. Hence, humans are sensitive to texture but are slow to adapt to it. Meissner's corpuscles bandwidth is 30-40Hz enabling fast adaptation to the stimulus. The Pacinian corpuscles bandwidth is unclear

as literature gives a variety of different range. A safe estimate one is 60 to 400 Hz, showing that human process touch and vibration with a fast adaptation to the stimuli [139].

2.2.1.2 Spatial and temporal resolution of the skin

Feeling granularity or surface topology is part of the interest in Haptic rendering. Spatial frequency or the capability of the human fingertip to identify variations of surface asperity is an important parameter for such applications. Research has shown that the fingertip can detect a dot of height 1-3 μm on a smooth glass surface [131]. Fingertip can identify two consecutive tactile feedbacks of 1ms duration separated by 5.5 ms at a single location [222]. Perception of stiffness is another important feature for Haptic interface. The stiffness threshold can determine when the user can feel an object surface and is use to categorise the type of object. For instance, under 25 N, objects are considered soft. In the literature, the Just-Noticeable-Difference (JND) of the proximal-interPhalangeal (PIP) and the MetaCarpal-Phalangeal (MCP) is about 2.5° [277]. The average JND for wrist, elbow and shoulder are respectively 2.0, 2.0 and 0.8°.

2.2.2 Human visual system (HVS)

Psychophysical research has proven that human eyes are relatively insensitive to information presented by colours compared to luminance e.g. brightness. In fact the retina is composed of 7×10^6 cones and 120×10^6 rods receptors. Cones are sensitive to colour and high spatial details whereas the latter to movement. Rods also operate at low light levels and are prone to colour insensitivity. Modern compression standards such as JPEG and MPEG are based on this feature of the HVS. A similar strategy can be used in haptic interfaces to render the VE in the display loop by using data redundancy or irrelevancy. Irrelevancy is linked to the perception of the VEs by human viewing. Redundancy is related to the statistical properties of data such as spatial correlation, spectral correlation e.g. RGB related to bit-depth, temporal correlation (figure 2.4) e.g. frame-to-frame or noise and the level of details (LOD) [12]. The relative sensitivity of the HVS decreases when the spatial frequency increases and is more sensitive in bright light than in dim light. Figure 2.4 highlights the higher sensitivity of the HVS for low temporal and spatial frequency. In applications where a head-mounted display (HMD) is used, often the field of view (FOV) is reduced due to the fact that it conceals the perception in the peripheral vision. Average HMDs possess a FOV between 70 to 120°. The feeling of immersion is increased by augmenting the FOV. However the resulting decrease in the resolution of the projected images will contribute to the sensation of motion sickness [243, 268].

2.2.3 Ergonomics, comfort and security

How exposure to VE affects the user is a very important concern. This topic is beyond the scope of the thesis but is an active topic of research involving, human performance efficiency, health and safety and social implication studies. Successful application of haptic sensing technologies should maximise the comfort of the users and avoid any risk of harm of injuries for the users. Maximizing comfort is certainly a key in improving the performance of the interface but is also a cause of fatigue which is an important issue, especially for exoskeletal devices.
There are several concerns about the impact of VE exposure for health and safety. Many effects have been observed. The most frequent macroscopic effect is cyber-sickness or motion sickness.

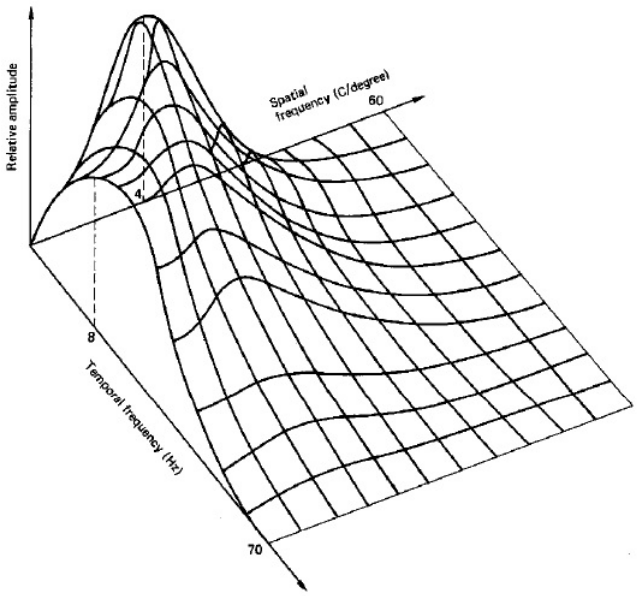


Figure 2.4: Temporal frequency response of HVS

The reduction of the FOV is one of the causes of such cyber sickness. It can also result from sensory conflicts between the visual and vestibular cues reaching the brain. Consequences in terms of perceptive effects can vary from headaches to nausea. Usually it is caused when the user perceives a time delay between his physical motion and the resulting movement in the VE.
Another issue is the one related to the visual cue and the use of HMD apparatus which imposes abnormal stress on the human vision much more than a regular display. Whether it uses a 3D effect or not, human eyes are closely coupled with the system which can result in strain of the human vision chain [243]. A poor adjustment of the HMD can generate flicker, glare or distortions. A much serious problem is the impact of HMD exposure on the potential increase of pre-existing ocular symptoms such as unstable or reduced vision acuity.
Other effects can result from the adaptation of human to their environment, in this case the VE immersive environment. After a long period of use, hangover of various time intervals can appear. The return to the real world require the brain to readjust certain of his perceptive cues. An uncomfortable feeling can be experienced during this period such as disorientation with reduced hand-eye coordination and vestibular disturbance. These wide range of effects are not so well understood and need to be studied in order to define safe regulation standards.

2.3 Haptic devices

From flight simulators, robotic gloves to surgery training tool, the family of haptic devices is very large and most of these device designs have been studied and explored to fit special particular applications. General survey of selected devices which are the most commonly encountered can be found in [53, 111, 283]. Devices are characterised by

- Their levels of degree of freedom (DOF),
- The force rendered,
- Maximum stiffness,

- Position resolution,
- The size of their workspaces and range of motion,
- The footprint,
- Their transparency.

Important features are maximum achievable stiffness and position resolution. The latter refers to the global system resolution and not just the encoder's resolution at the motor which ignores lost motion and compliance in the mechanism. Backdrive static friction is a very important specification for impedance-type haptic devices. Friction is caused by mechanical effects in bearings, cables, and motor brushes. Admittance device have zero static friction.
Another important attribute is the perceived mass(inertia) at the user end. Having a low perceived mass mostly determines how the device feels when moving rapidly through free space. Perceived mass is much important during interaction with VE when feeling contact events with rigid and flexible objects.
Devices can be ground or body-based (glove, suits) as shown on figure 2.5. The issue with body-based devices is the total absence of space tracking. To obtain the position of the device in the virtual space these systems have to be coupled with an optic tracking or inertial system. Another issue can be discomfort due to the weight of the apparatus. Haptic devices can be divided

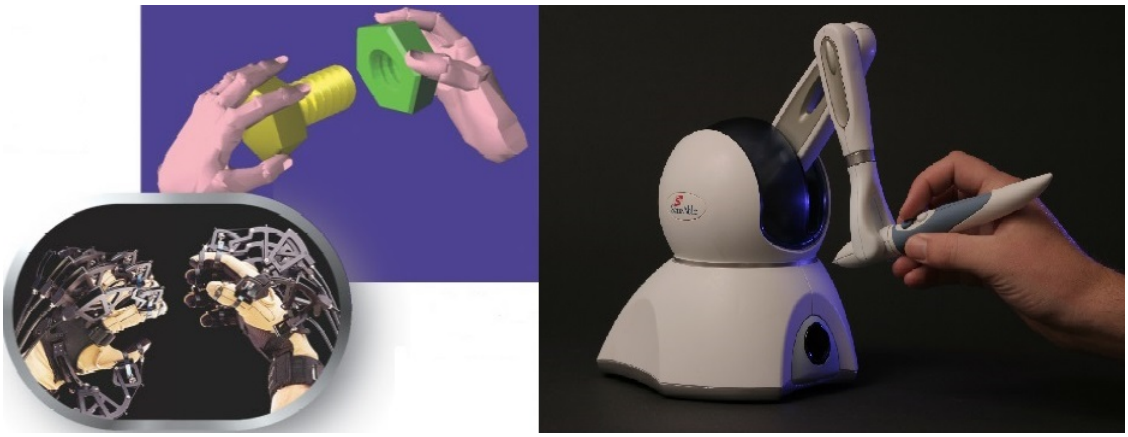


Figure 2.5: Body-based device haptic gloves (left) vs ground-based Omni device (right)

into two different classes [283]: Impedance or admittance controlled devices. Table 2.2 shows a comparison of their different features.

Control type	Advantages	Drawbacks
Impedance	Light weight, small. Can render mass and light friction	No able to render: high force, high mass, stiffness.
Admittance	Can render high stiffness and absolute zero friction High transparency for motion feeling	Difficult to render low mass Big footprint

Table 2.2: Comparison of control types for haptic devices

The haptic interface used in our project is the PHANTOM®OMNI device manufactured by Sensable [253] technologies (figure 2.5). This device belong to the "pen" type where the force and torque are rendered by a 6 degrees of freedom (DOFs) robotic arm controlled by electrical motors. The pen or stylus is located at the end of the arm allowing the user to interact with virtual environment. It allow a much realistic interaction than the simple 3DOF which is limited to point-object interaction with no torque rendered [284]. The primary reason for the choice of this device is its low cost, ease of use and ergonomics. Impedance device is suitable for our application because high force feedback is not needed for the design process. The position accuracy provided by such Haptic tools is ≈ 0.055 mm which sufficient for our current needs. The workspace of the interface is 160 width $\times$120 high$\times$120 depth mm. The maximum force that can be rendered is 3.3 N with a frequency of 1Khz. Back-drivable-friction is less 0.26 N. This is clearly not optimal for a tool dedicated to CAD of MEMS but is sufficient for the thermal prospecting application we target in this thesis. It allows the interchangeability of handles and adding a 7^{th} grasping DOF, to suited different applications.

Haptic devices are usually coupled with immersive vision apparatus such as 3D screen, or 3D goggle or helmets. They can also be used within highly immersive environments such as the CAVE (figure 2.6) where walls and floors are display systems in order to completely immerse the user in a virtual world. There is some research devoted to the use of such devices for immersive collaborative work. Most of the applications revolve around scientific data visualization [195]. At

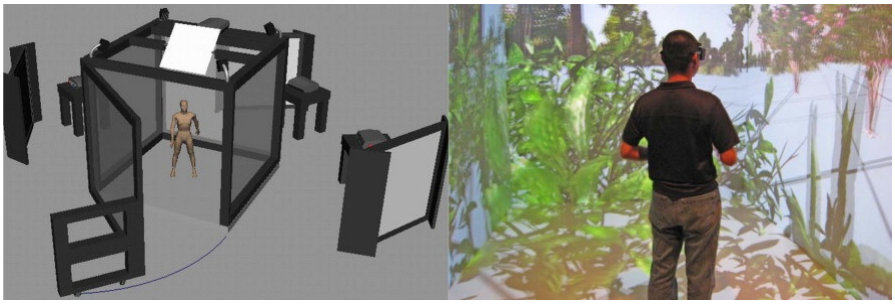


Figure 2.6: The CAVE

the nano and micro scales, the mechanical behaviour of objects is no longer dominated by gravity and friction. At short range, forces including electrostatic, capillary, and Van Der Waals forces are predominant [89]. As a result, the physics of the micro/nanoscale differs completely from that of the macro-scale and are not accessible to human sensory and motor capabilities. Haptic devices can be employed to magnify interaction between objects at a macro-scale at which humans are naturally effective. Most haptic devices fail to match the human sensorimotor capacity and thus act as an obstacle between the hand and the phenomenon that could be accessed. Conventional interfaces are subjected to inherent friction and/or high inertia that affect the interface transparency, dynamic range, and bandwidth. At the micro/nano scales, three issues arise in term of interface design [110].

In past years, many haptic interfaces have been proposed to deal with interaction with small objects. They enhance human interaction capabilities at the micro-scale using either electrodynamic transducers [119, 127], or single stage systems based on conventional robotic architectures, see [87], among several other examples. Such interfaces are subjected to inherent friction and high inertia, which contribute spurious forces that mask the mechanical signals to be felt by the users. With single stage systems, sacrifices must be made regarding transparency, dynamic range, or the

bandwidth in favour of other factors such as maximum force.
Dual-stage haptic device [185] designs are suitable to address the needs of interactions at the micro/nano scales. A large, proximal motor provides power and the small distal motor reproduces the transient forces. The two stages are connected to each other by a passive viscous coupler based on Foucault currents. Such a device allows good transparency and should be preferred for final implementation for haptic MEMS design.

2.4 General haptic architecture

The main challenge of a computer-haptic application is the real-time constraint. A large body of literature details the requirements and the basic architecture for a haptic system [20, 38, 114, 175, 244]. The physical properties of the model have to be updated at the rate of 1KHz to ensure a continuous haptic feedback. Figure 2.7 shows how haptic systems can be divided in two loops. The haptic loop needs to be updated at the rate of 1 KHz in order to provide continuous force feedback perception by the user over time and avoid discontinuity or artefacts. The graphics loop is updated at the rate of 30 Hz minimum to maintain a continuous display of the scene. Both

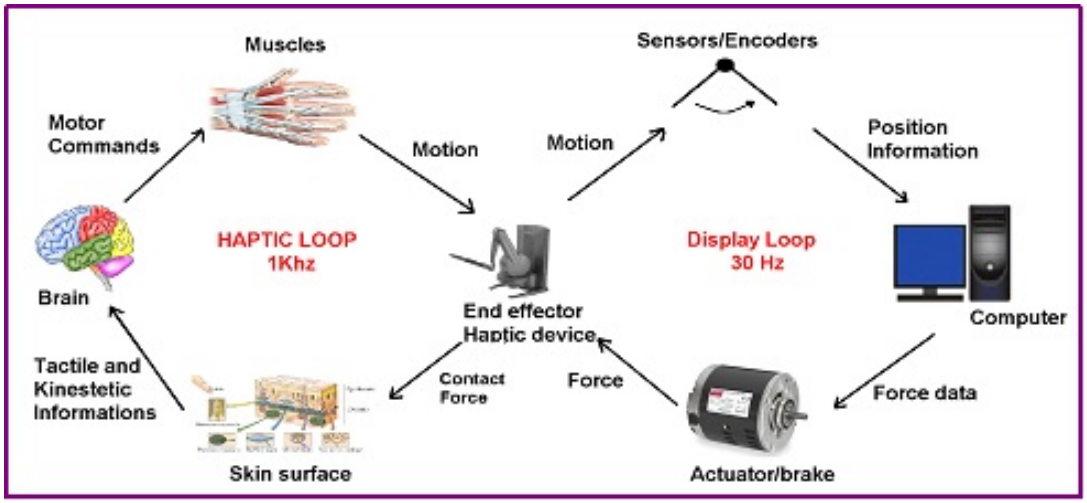


Figure 2.7: Haptic system architecture

loops are running concurrently when the user is moving the haptic interface e.g. the stylus of the phantom device in our case. The position of the stylus is read continuously by the haptic interface and used within the collision detection algorithm to access the model information from the database. When collision is detected the database is updated if necessary and physical forces response, physics of model computed. Information is then converted by the haptic interface to produce force feedback. Figure 2.8 describes the typical software-hardware architecture for a haptic application. The computation load necessary for rendering the scene in 3D represents a significant part of the algorithm. It makes the integration of the collision detection more difficult. One of the challenges of haptic force rendering is not only to assure a continuity of the feedback over time but also through space. This contributes to the complexity of collision detection.

2.5 Geometry

The first step in all design or engineering simulations is the creation of the geometry describing the shape of the region of the problem to be analysed. Computational geometry deals with rep-

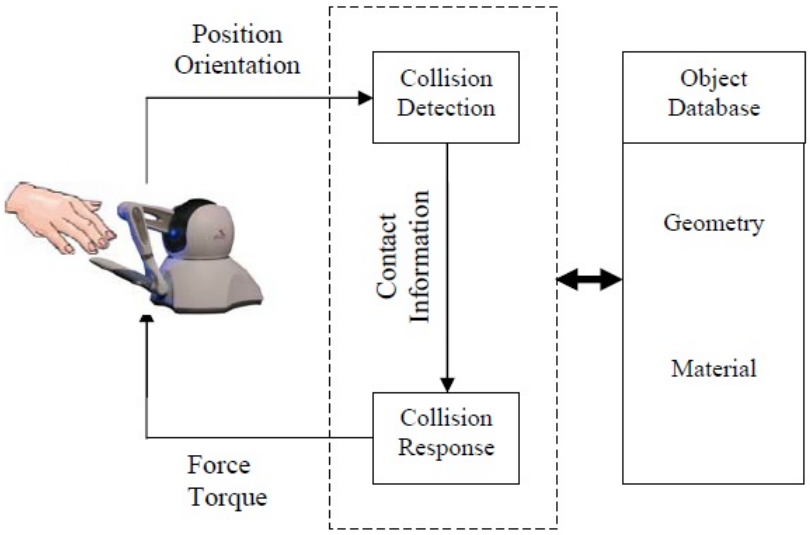


Figure 2.8: Haptic interaction [240]

resenting geometry through the use of computer resources [35, 78, 96, 149, 232]. Rendering the geometry of an 3D object requires the creation and storage of different elements (figure 2.9) such as

- Vertices represent a position along with colour, normal vector and texture coordinates.
- Edges are the connection between two vertices
- Faces are surfaces defined and enclosed by four edges.
- Polygon or volume defines a set of faces.

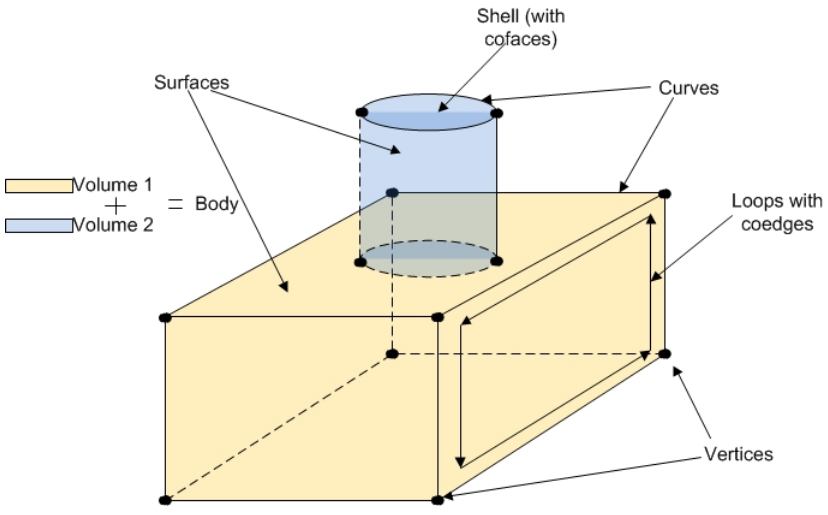


Figure 2.9: Geometry decomposition

Methods for 3D object model in VE are categorized into polygonal and non-polygonal ones (figure 2.10). Among non-polygonal model, constructive solid geometry (CSG) and surfaces geometry are the most used. In CSG, objects are defined as a combination of primitives shapes such as bricks, spheres and cylinders such as rectangular blocks and cylinders associated with operators [233]. Usually in CSG, the combination is expressed as a "CSG" tree in which the non-terminal

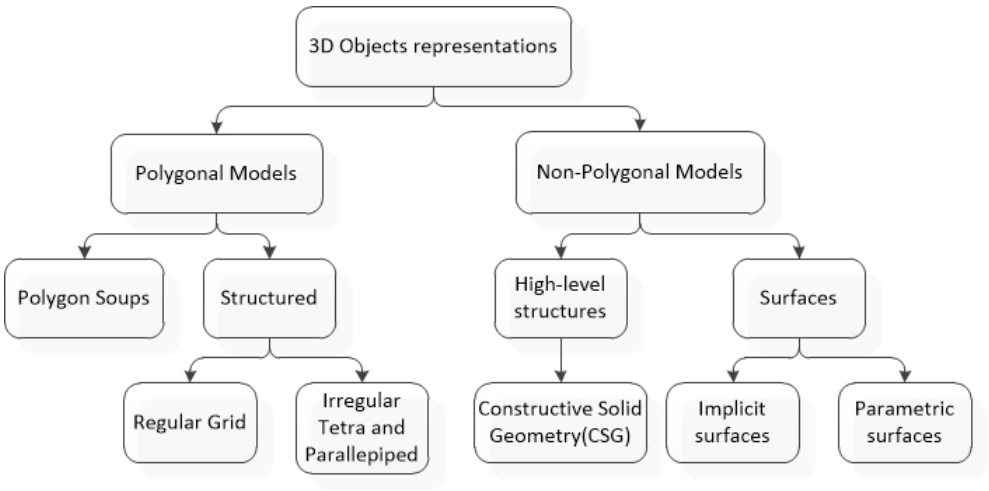


Figure 2.10: 3D object representation methods

nodes are operators and leaves are primitives [149, 150, 225]. Rendering 3D object can be done by convex envelop of points, curve and surface with Non-uniform rational B-spline (NURBS). Procedures for creating such geometries in computer graphics follow either a bottoms-up approach where vertices are created first, then connected with the edges to form faces then volumes or the top-down approach.

Mathematically a polygonal mesh may be considered as an unstructured grid or undirected graph, with additional properties of geometry, shape and topology. These geometry elements can be regrouped into different object entities created and manage by CAD softwares [213]. Usually, mesh connectivity structures do not require storage of all these objects. Mesh generator for 2D or 3D computer graphics application can generate some objects structure from others, like polygons from vertices, edge and faces. Most of the mesh generators or renderers nowadays can construct polygons from a set of triangle meshes or vice-versa.

The choice of the modelling methods for 3D virtual MEMS objects in haptic application depends highly on the application. Required accuracy, complexity of the collision detection method, need for dynamic deformable objects and finally computational cost resulting from data storage are the primary criteria of choice. How 3D virtual objects are modelled within the VE have a great impact on the collision detection algorithm efficiency [116].

For simple models with no deformations, CSG can be used. Haptic interaction can be associated with geometric primitive shapes. In the case of deformations or complex shapes, mesh is preferred to render complex interactions.

2.6 Haptic rendering

Haptic rendering is the process of displaying and rendering forces in response to user interactions with the VE. The process of haptic rendering can be separated in two parts: a collision detection and a collision response algorithms as shown on figure 2.8.

2.6.1 Type of haptic interaction

The first parameter affecting the computational cost of the haptic rendering algorithm is the type of interactions with 3D objects allowed by the system. In our case it corresponds to the model used

to represent the stylus in the VRE. There is globally 3 types of possible models (figure 2.11) [20]. The simplest model is the point-based or point-object interaction where only the end(virtual point)

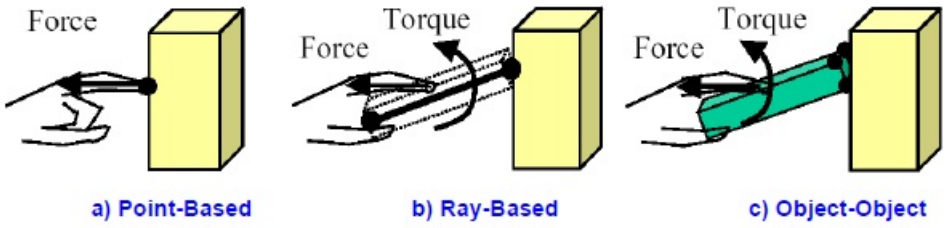


Figure 2.11: Different model for stylus interaction with 3D components in the VE.

of the stylus is modelled in the VE and can interact with MEMS virtual components [114]. During each cycle of the haptic loop, the collision detection algorithm will check if this virtual point is inside any virtual object. The collision detection is the fastest for this type of model as the check is done with only one point resulting in minimum data manipulation through the pipeline interface. If collision is detected, the penetration depth is computed using the difference between the current virtual point position of the probe and the projected point by the normal on the penetrated surface. We will use this type of model for our test as it allows point application for heat source.
The second type of interaction is the ray-based model which represents the probe as an oriented line [19, 116]. Another solution is to represent the probe as a set of connected segment [115]. This method of interaction is slower than the point-based technique. Ray based models introduce segment-edge and line segment-polygon interaction. The last method is to model the probe as an object in the VE. This method is referred to as object-object and produces complex interactions. Both ray-based and object-object models allow collisions with more than a single object and multiple layers at a time. In this case the haptic rendering and collision detection is much more computational expensive. Depending on the type of application, we have to make a trade-off between the realism of the interaction with the VE and computational cost.

2.6.2 Collision detection methods

The role of collision detection (CD) is to detect any geometric contact between the user virtual end and the component present in the VE [20]. Collision detection methods can be classified according the virtual object modelling methods used

- The surface representation based model or volumetric based model. The most popular methods are the constrained-based method and the implicit surface based method.
- Volumetric methods available are Bounding box, hybrid hierarchical representation (H-collide) and voxel based methods.

2.6.2.1 Penalty methods

The penalty method was one of the earliest collision detection algorithms. It is based on Point-Object model. In this method the volume is subdivided and each sub-volume is associated with a surface (figure 2.12) [114]. A force vector is associated to each sub-volume with its length proportional to the interpenetration. In this model the force feedback is determined directly from the penetration depth. This method works well for simple shapes but presents several limitations [238].

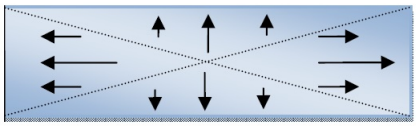


Figure 2.12: Illustration of the penalty method

The first one is the force discontinuity that can arise when the probe is crossing the boundaries of sub-volumes (figure 2.13, a). A second issue with this method is the lack of locality. As shown

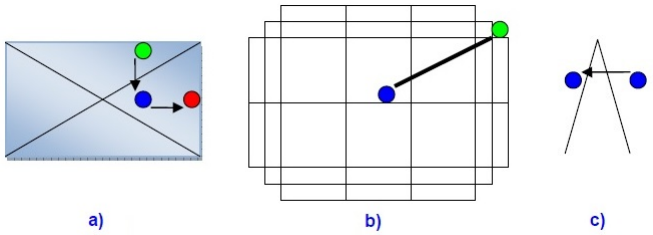


Figure 2.13: Penalty issues: a) force discontinuity l; b) lack of locality ; c) pop-thru effect

on figure 2.13.b, it can be difficult to determine which exterior surface should be considered when the stylus is within the object. The last problem that can arise is the pop-thru effect that is caused by an insufficient internal volume of the penetrated object (figure 2.13,c).

2.6.2.2 Constrained based method

Similarly to the penalty method, constrained based methods are a Point-Object type of interaction and are based on the surface of the geometry. In this method, the idea is to keep track of the stylus, also called the Haptic Interface Point (HIP), during the haptic interaction by storing its path history. This allows to overcome the issue generated by the penalty method. The method consists in letting the HIP penetrate the object and constrain the virtual location, also called virtual proxy [238] or Ideal Haptic Interface Point (IHIP), of the haptic interface to remain at the closest point on the surface (figure 2.14). In our case the HIP is the stylus's end point. The position of the IHIP is computed using Lagrange multipliers [308]. When the HIP collides with an object in the VE, the

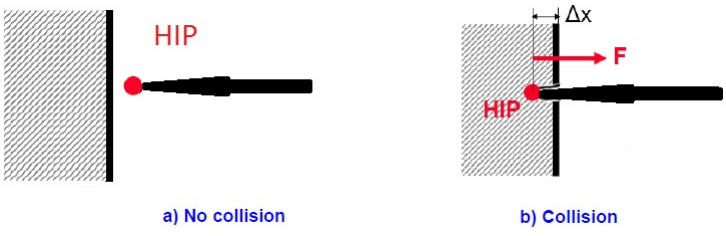


Figure 2.14: Constrained-based method

force feedback is generated using Hooke's law $F = k\Delta x$ with k being the stiffness coefficient and Δx the penetration depth. Low values of stiffness coefficient will generate perception of soft objects whereas large coefficients will generate perception of rigid surface. For realism purpose, a damping coefficient b is usually added and such as $F = k\Delta x + bv$. Big stiffness coefficient often generates unstable interactions and undesirable vibrations. The force at the entry is large and

when the HIP is moving out of the body, the damping coefficient becomes 0 to avoid the stiction effect. The procedure is illustrated in figure 2.15. In the case of free space with no collisions,

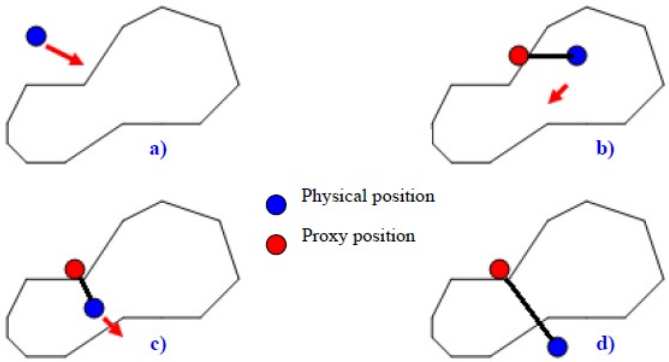


Figure 2.15: Constrained-based method principle

the real position of the stylus (blue) and the IHIP (red) are the same (figure 2.15.a). When the HIP penetrates an object in the VE, the IHIP remains on the surface (figure 2.15 b and c). In the event the stylus is passing through the object (figure 2.15 d), the IHIP remains at the surface entry point. At each iteration a check is executed to determine if the HIP is still inside the object by constructing the vector passing by the IHIP to HIP [308]. If the dot product is negative then the HIP is outside of the object. On the other hand if the product is positive, the HIP is still inside and the feedback force can be generated using both the penetration depth and the constructed vector.

2.6.2.3 Implicit surface

Implicit surface representation of the objects in VE can be used for collision detection [135, 241]. The external surface is described by a function of the form $S = \{p(x, y, z) \in \mathbb{R}^3 | f(p) = 0\}$, where S is the surface of the object and p the location of the Stylus. This provides an efficient way to detect whether the stylus is inside or outside the surface similarly as would do a distance signed function. If $S(p) < 0$, the stylus location p is within the object otherwise if $S(p) > 0$ or $S(p) = 0$ the stylus end point in VE is located respectively on outside the surface and on the surface. Surface normal n is computed using the Gradient of the implicit function

$$n = \frac{\nabla f}{\|\nabla f\|} \tag{2.1}$$

One problem with implicit surfaces is the difficulty in determining which point on the surface should be used to render force feedback during collision. Usually in order to increase the smoothness of the surface, the normal is computed by interpolating the gradient of its neighbours around the contact point. Projection technique can be used to interpolate an approximate point on the surface [241].

2.6.2.4 Bounding volumes

Spatial partitioning method i.e. Bounding volumes is an efficient and simple method for simple VE objects. The method uses either an Axis aligned Bounding Box (AABB) or Oriented Bounding box (OBB)(figure 2.16). The AABB is built by simply taking the maximum and minimum coordinates of the objects in the VE along the x, y, z coordinates to create an encapsulating Bounding

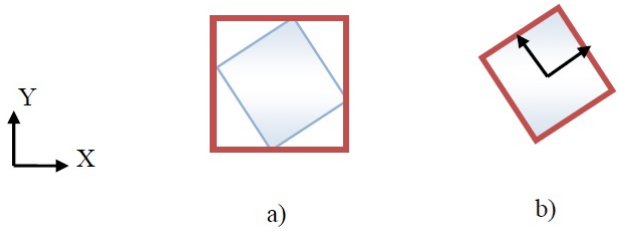


Figure 2.16: AABB (a) vs OBB (b)

Box (BB). It can be used to determined collision between objects in VE. A collision is detected if the three orthogonal projections of the BB of two objects overlap in the xy, yz and zx planes. In our case, we consider the stylus as point $P(x, y, z)$. First the algorithm computes the BB of the close objects and then checks if the point P is simply inside the box. If collision is detected, the algorithm proceeds to local collision detection. OBB is also a rectangular box is oriented in such a way that it captures the shape of the object more efficiently. Figure 2.16 demonstrates how OBB are superior in enclosing more accurately shapes. AABB and OBB are used with AABBTree and OBBTree for hierarchical structures for the VE scene. [98] demonstrated the use of OBBTree algorithm for VE with complex objects composed of large number of polygons and even polygon soup. It also permits close proximity of objects and multiple contacts with good accuracy. It uses an axial projection of the BB onto axis in space and compares the interval of each box on the axis to determine potential collision if an overlap occurs. Ho and Basdogan [114] proposed a neighbourhood connectivity hierarchical tree database for each object in the VE, to be used to performs local searches of subsequent contact. Other type of Trees i.e. hierarchical structure such as k-d trees, octrees [242] or sphere trees [121] can be used. BB is an efficient technique to perform collision detection between rigid objects but has proven to be rather difficult for non-rigid deformable objects.

2.6.2.5 Voxel-based

This method partitions the VE into voxels and checks its content at the stylus position. Grid partitioning for constant voxel's size or octree for different voxel's sizes can be used [9]. The surface of dynamics objects are represented by a collection of points with associated normal oriented inwards. These are called point shell. The static objects are represented by their voxmap and the numbers of the voxels occupied. 2 bits are used to identify the voxel status. There are four different states possible for each voxel: free space, proximity, surface and interior (figure 2.17). For collision detection, the algorithm updates the contents of the voxels accordingly to the location of the point shell, and checks if it tries to write into a voxmap which already contains the voxel of another object. McNeely [181] applied a voxel based algorithm for collision detection and haptic rendering for a single dynamic rigid object in the VE.

2.6.2.6 Hybrid hierarchical representation(H-Collide)

The hybrid hierarchical method i.e. H-Collide [100] is a framework for fast and accurate collision detection for haptic interaction. It is a two stage method. First it computes a hybrid hierarchical representation of objects during a pre-rendering stage (offline) followed by an on-line computation. H-Collide first processes a spatial decomposition of the workspace into uniform grids or cells,

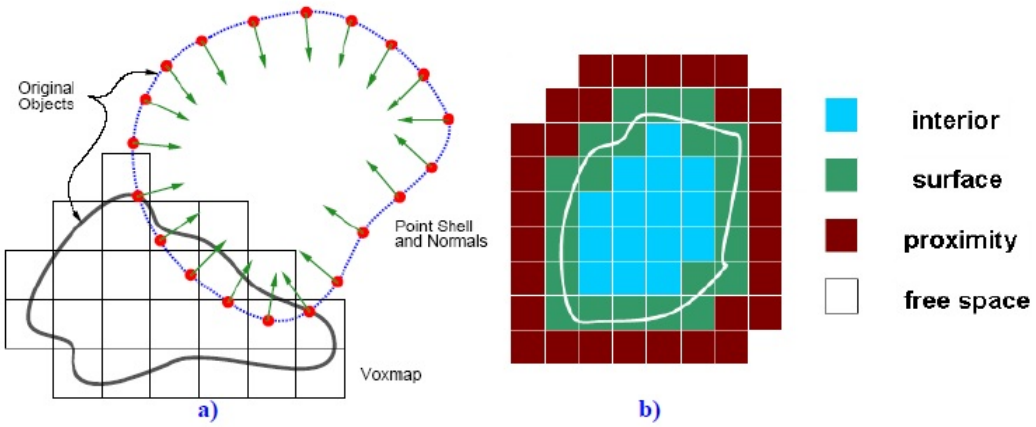


Figure 2.17: Point shell colliding with voxmap (a); voxel map sampling (b)

implemented as a hash table to efficiently deal with large storage requirements. At run time, the algorithm quickly finds the cell containing the path swept out by the probe.
It start off by building a Bounding Volume Hierarchy Based on OBBTrees. An OBBTree is a bounding volume hierarchy and each node of the hierarchy corresponds to a tight-fitting OBB. For each cell consisting of a subset of the polygons of the model, we precompute an OBBTree. At run time, most of the computation time is spent on finding collisions between an OBBTree and the path swept out by the tip of the probe between two successive time steps. An optimized overlap test between a line segment and an OBB has been developed [100], taking as few as 6 operations and only 36 arithmetic operations in the worst case (not including the cost of transformation). The disadvantages of this method are the off-line pre-computation and the consequent storage requirement. These make the H-collide not adapted for deformable objects.

2.6.2.7 I-Collide and Voronoi-clip (V-clip)

I-Collide and V-clip are similar techniques which calculate the Euclidean distance of the closest pair of points between two moving convex polyhedral based on Voronoi regions [65,155]. Voronoi regions are partition space of the VEs outside the polyhedron characterised by a set of points having a common feature with the closer polyhedron. A Voronoi diagram of a polyhedron is a set of Voronoi regions of that polyhedron. These techniques exploit the fact that between iterations, objects in the VE change very little. This is called the temporal geometric coherence. The algorithm suit large scale environment well and its efficiency can be increased by using the Sweep and Prune [65] which reduce the number of pair-wise collision tests by eliminating pairs which are unlikely to collide during the next iteration.
V-Clip algorithm are an improved I-Collide, which return either the closest points between objects and the distance between them or the penetration depth in the case Objects are interpenetrating [186]. V-clip, unlike I-Collide does not handle the overlapping objects. V-clip performs well for rigid objects but is not suitable for deformable objects since the algorithm is adapted for convex polyhedral objects. It also performs less arithmetic divisions, which reduces its computational cost. Methods based on incremental algorithm working on Voronoi regions for convex primitives have been developed [101]. By using a look-up table to define which vertex of a primitive is near a given direction, the collision detection can be operated faster. However these techniques have difficulties meeting the update rate criteria for haptic.

2.6.2.8 Bubble technique

Another technique worth mentioning is the "Bubble" technique which has been developed for its ability to interact with large VEs. It is based on a hybrid position/rate control which enables both accurate interaction and coarse positioning in a large VE. The haptic workspace is displayed visually using a semi-transparent sphere that surrounds the manipulated cursor. When the stylus is located inside the sphere, its motion is controlled for fine positioning. The user may also "feel" the inner surface of the bubble, since the spherical workspace is "haptically" displayed by applying an elastic force feedback when crossing the surface of the bubble.

2.6.2.9 Overview of the Collision methods

An overview of the different methods is given in tab 2.3. Considering our application needs for MEMS design with deformable objects under constrain and the fact that we will not work with multiple objects in the VE, we will use the available constrained-based methods implemented within the OpenHaptic framework [254].

Type	Method	Advantages	Drawbacks
Surface representation	Constrained based	Simple method. Solve issues of penalty method.	Computation time increase with the VE's complexity.
	Implicit surface	Ideal to model general object and the property of the inside-outside interaction.	Difficulty in determining the point on the surface to render the feedback. Results in extra computational effort.
	I-Collide/V-Clip	Fast CD	Not suitable for deformable object. Difficulty to meet the update rate requirement.
Volumetric methods	Penalty methods	Simple	Force discontinuity. Lack of locality. Pop through. No past history.
	H-Collide	Robust	Off-line pre-computation. High storage requirement. Not suitable for deformable objects.
	Voxel	Good for modelling dynamic and deformable objects	Computationally expensive for complex environments.
	Bounding volumes	possible for dynamics objects CD. Multiple contacts.	Difficulty for CD and and intersection of Non-rigid objects.

Table 2.3: Overview of collision detection spatial haptic rendering methods

2.6.3 Surface properties rendering

A very important concept for haptic VE immersive feeling is the rendering of surface properties. It significantly increases the feeling of realism by the user. In the following we describe some of the most common techniques used to render material surface properties.

2.6.3.1 Force shading

Force shading is similar to Phong or vertex shading used to smooth polygonal surfaces in computer graphics. The surface normals are interpolated using neighbouring vertices normals [19]. Figure 2.18 illustrates the principle of the shading. In the case where the point of contact is a primitive

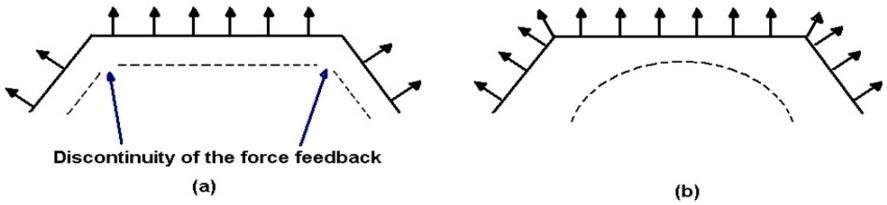


Figure 2.18: Comparison of haptic methods: with discontinuities (a) constraint based method, (b) force shading

vertex, the normal at the contact point is the one of the vertices. In the case of contact with a line primitive, the normal is the average of its two neighbour vertices. In the case of surface, average weighting methods is used. For triangle mesh surfaces, the triangle collided is sub-divided in 3 sub-triangles as shown in figure 2.19. The surface normal N_s at the contact point is computed

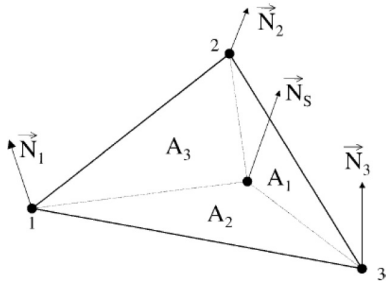


Figure 2.19: Averaged surface normal N_s

using the weighted normal N_i at each vertices of the triangle by the area of each sub-triangles.

$$N_s = \frac{\sum_{i=1}^{3} A_i N_i}{\sum_{i=1}^{3} A_i} \tag{2.2}$$

A problem with shading force with that it can create an undesired sensation of roundness.

2.6.3.2 Contact impedance

When the stylus is in contact with an object's surface in the VE, the generated force feedback F direction is the normal of the surface and is characterised by the combination of a spring force and viscous damping force such as $F = k\Delta x + bv$. K defines the stiffness of the material at the contact point, Δ is the difference between the coordinate of the stylus and the surface contact point (SCP). b is a damping coefficient and v the velocity of the stylus end.

2.6.3.3 surface friction

When the user stylus is in contact with an object's surface and moves along it, surface friction is rendered by generating an opposed tangential force [240]. This type of friction is called Coulomb friction and it can be either static or dynamic (kinetic friction). It is usually done by adding a proportional lateral force vector F_f to the normal force vector n such as $F_f = \mu n$. Here μ is the friction coefficient. In practice the static friction resists the movement of the stylus when in contact with the object surface. When the force applied by the stylus overpasses a predefined threshold, the surface friction rendered will be slightly decreased over a short period and will reach a constant values as long as the stylus keeps moving along the surface.

2.6.4 The level of detail

In haptic technologies, robust rendering of an accurate continuous force feedback in real-time is a difficult task due to the large amount of data processed by the haptic loop at a frequency of 1KHz imposed by the human sensory kinesthesia chain. The understanding of human sensory information is important in order to address them accurately and efficiently, but it can also be used at our advantage to reduce the quantity of data that has to be processed.
In audio virtual rendering, psychoacoustic can be used to compress and treat audio signal in order to deal with a reduced amount of data and keep the level or fidelity high [39, 205]. Similarly we have seen that the human visual perception is limited in accuracy. Reduction of the computational cost can be done by controlling the level of detail (LOD) of an object represented in the VE. By considering only shape of objects, LODs can be used to determine the distance between them [303]. Controlling the LODs can be complex for VE composed of 3D thousand or millions polygons objects. The same object is basically rendered with different numbers of polygons (figure 2.20). There is a huge benefit in terms of computational cost in switching LOD model, however

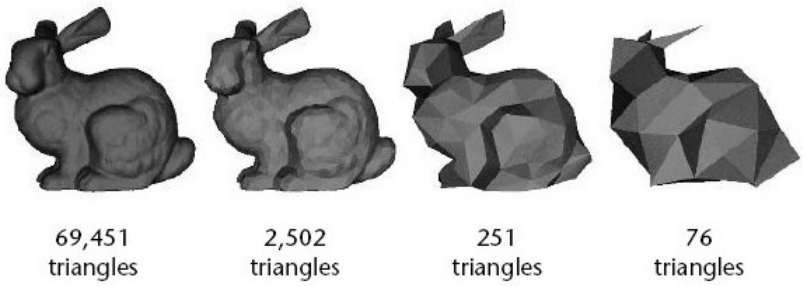


Figure 2.20: Model with different LODs

LOD management techniques require good effort in programming. There are 3 different classes of techniques that can be used for controlling and matching the resolutions of the mesh according user's view. The LOD is a function of the location of the 3D object in the VE's. The same object moving will see is LOD changing as he moves in the foreground or in the background of the VE scene. A simple method is to decrease the LOD as the object gets further away.
Another technique is to have the LOD function of the velocity of moving object. This technique controls the LOD by substituting a coarser model to the fine one. This reduces the haptic rendering computational cost. When the object is moving fast or at a distance, the LOD is decreased [303]. One needs to pay attention to the transition in switching models to avoid perceiving jump in the

LOD, thereby creating artefacts for the user. Mapping function can be used to update the model in function of its position on the VE.
The last technique is to make the LOD a function in the regions of the object interacting with the stylus. One can subdivide the surface of the object where the collision is occurring with the stylus and coarsen the considered object mesh. Such a technique has been used for soft object simulations where intensive computation is required [284, 304]. A coarser mesh is used for detecting the contact and a finer mesh is adopted for regions colliding with other objects or the haptic probe.

2.7 Conclusion

The second chapter focused on the description of the haptic technologies and the motivation for integrating such interfaces in MEMS design and simulation. We have reviewed the general requirement for efficient haptic interfaces and human physiology criteria which have to be taken into account for haptic feedback perception. We also cover the different haptic rendering and CD methods.
We have shown the potential of haptic sensing based design to reduce the cost and time of design process by avoiding the long and expensive process of prototyping. Haptic sensing can help designers to quickly and better grasp the dominant roles played by forces at micro-scale. It can also assist users in their design choices, using haptic channels to constrained manipulation and assembly of parts and integrating micro-scale effect such as Van der Waals force.
Another potential benefit from haptic environment is for analysing complex sets of data at the design stage of the product. The formulation of MEMS sensor technology quite often leads to multi-physics problem. For instance, when thermal problem is coupled with other physics problem such a mechanical, electromagnetic, fluidic or acoustic then the visualization and the extraction of data can be difficult as the colour chart is usually the main indicator for several field data such stress, temperature, pressure, magnetic, pressure flow. Haptic creates another channel where other form of feedback such as vibration, stiction, attraction force can be used to convey more data information. Hence, it can be used for haptic MEMS thermal management problems to provide direct feedback on the modifications produced by thermal effects involved in the design process.

Chapter 3 Implementation of haptic tool demonstrator for thermal management of MEMS design

3.1 OpenHaptic framework

The PHANTOM Omni device is interfaced to a PC via a PCI controller card. The communication between the PCI card and the computer is handled by a dedicated device driver. Its function is of primary importance as it maintains the 1KHz update of the servo-motor in order to ensure the stability state of the device. There are a few software development kits (SDKs) available for rendering tactile-feedback.

Many of these SDKs or application programming interfaces (APIs) are C++ based toolkits. Their purpose is to make the creation of virtual objects in VREs and the interaction with them easier. For our application we are using the SensAble API OpenHaptic [254, 255]. OpenHaptic SDK embeds the Haptic device application programming interface (HDAPI), the Haptic library API (HLAPI) the PHANTOM device drivers (PDD) and some utilities. It provides useful libraries in order to command and manage communication with the haptic device. The HDAPI is a low-level direct machine control API, which provides routines to manage the thread's priority and frequency for communicating with the device. The core of the API and Driver is C/C++ based.

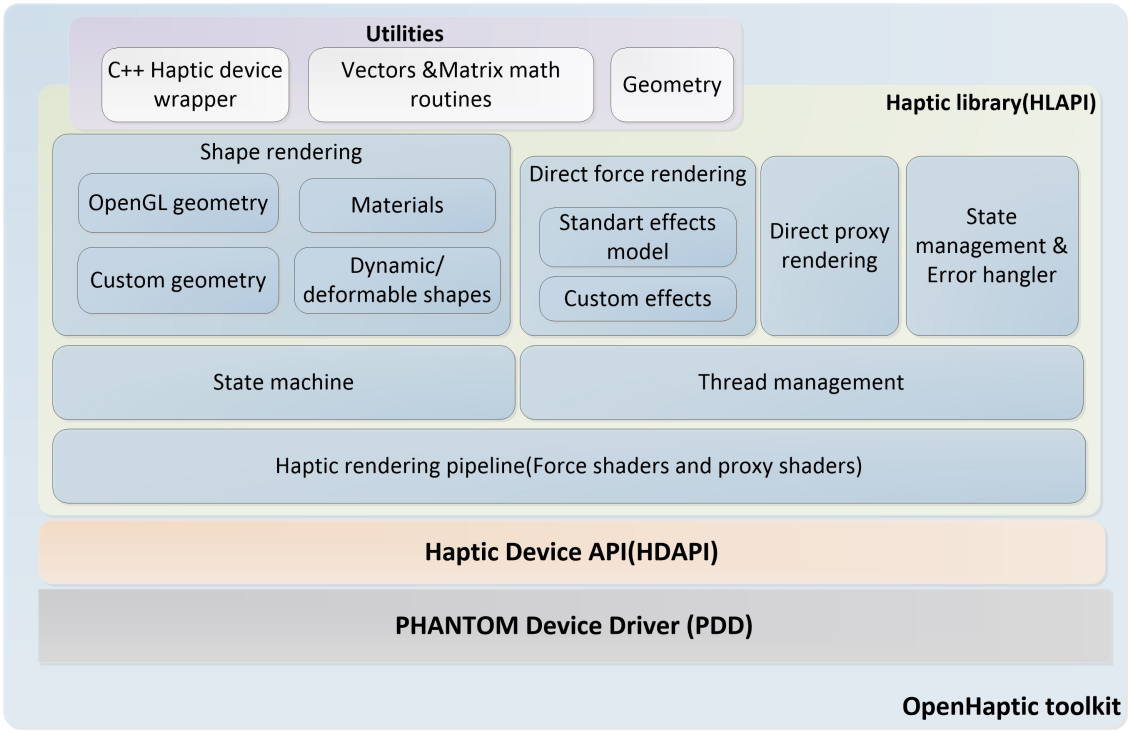


Figure 3.1: OpenHaptic layout

The HDAPI allows access to control encoder values and DAC signal values. It has components to manage the state system, send forces and set parameters. The scheduler part manages the

frequency, and high priority thread to render forces and retrieve information from the device. It also allows the addition of operations in the servo loop thread. Table 3.1 shows the major functionalities provided by the HDAPI.

Get device characteristics	Workspace dimensions, I/O DOF Max stiffness, max force, max velocity Motor temperature Calibration capabilities
Get device state	Position, orientation, velocity Coordinate spaces, raw data I/O from encoder Buttons
Set device	Force, torque Motor command
Enable/disable	Force output, clamping ,ramping Error report Checks: max force and velocity
Device Calibration	Auto/manual calibration
Scheduler	Synchronous/asynchronous callbacks Scheduling call backs Haptic servo rate
Error report	Function, device Force Haptic rendering scheduler
Utilities	Workspace to camera mapping Vector and matrix math objects Geomtric routines C++ Haptic device wrapper

Table 3.1: HDAPI overview of functionality

The HLAPI is built on top of the HDAPI and provides high level control and haptic rendering command (figure 3.1) such as functions for geometry interaction. It provides routines off the shelf to interact within the VE with the haptic device. However this command does not allow much flexibility in terms of the shapes available for interaction as well as the type of haptic rendering. The HLAPI also allows us to use OpenGl command to create haptic rendering based on geometric primitives. This greatly simplifies the implementation. It also incorporates all the callback functions to manage buttons action and haptic end proxy collision. Control routines are also available to control the force produced by the haptic Omni device. Table 3.2 shows an overview of the functionality of the HLAPI. The HLAPI also allows us to modify the force feedback of the haptic device. This is particularly important for our application using custom effect for haptic texturing (see section 3.4). It is also possible to integrate physics and CD.

Tests have been carried on Pentium 4 Ht 3.2 Ghz using windows OS. All haptic CAD tool tests have been coded using C/C++ language and graphic library OPENGL [99] with GLUT (GL Utility Toolkit) [134, 224] which provides solution for windows creation and management on Windows OS platform. Figure 3.2 shows the interface using GLUT and VE environment build using OPENGL/HDAPI.

Shapes	OpenGL primitives(vertex, lines ,polygons) Custom
Force effects	Constant Viscosity, friction Spring Custom
Interaction type	Point Contact/Constrain
Surface properties	Friction Stiffness /damping Front,back faces
Dynamics	Allow the integration of physics or dynamics and CD engines
Deformable objects	Integration of libraries or engines for deformation
Events handler	Mouse, keyboard Haptic device: Touch,untouch, stylus roller switch Motion of stylus Calibration

Table 3.2: HLAPI overview of functionality

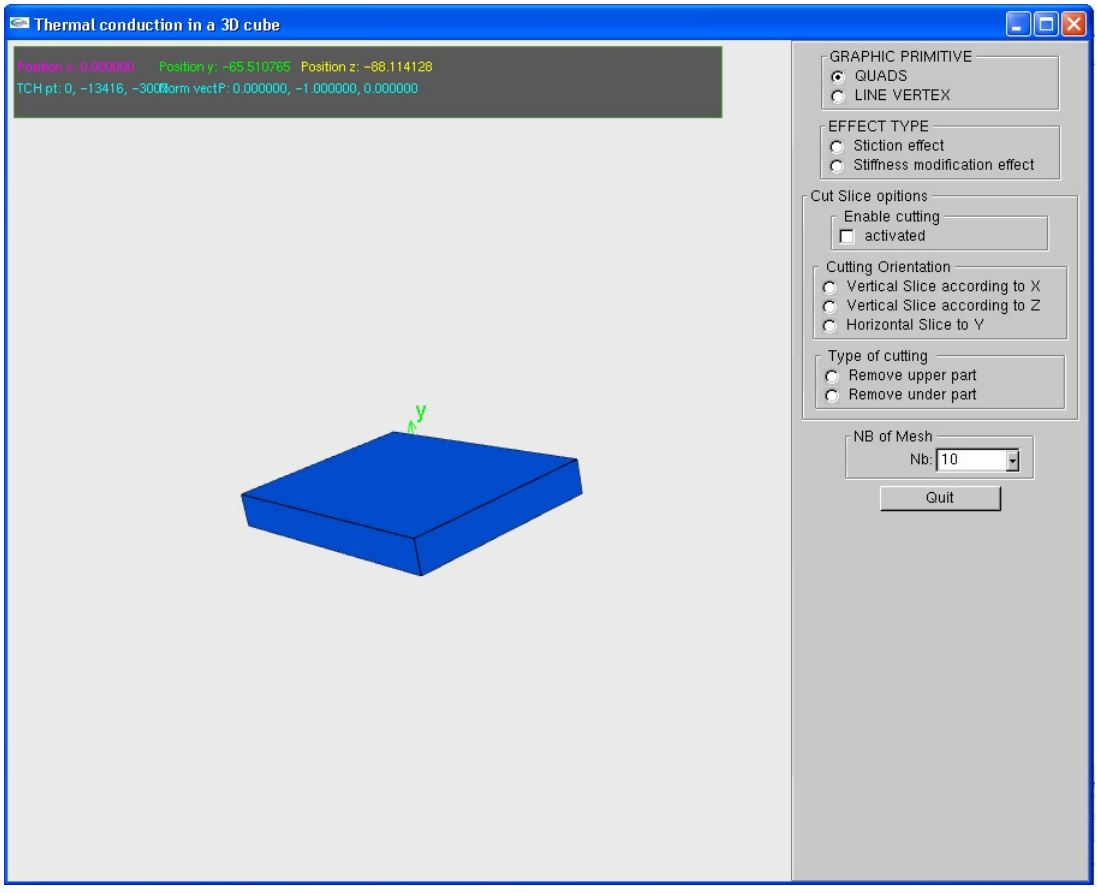


Figure 3.2: Haptic graphical interface

3.2 Modelling technique for objects physics

In our desire to develop a haptic tool for the thermal management of MEMS during their design, the simulation of the model's physics is a key feature. In haptic, methods to model the physics of component and more specifically geometric deformations can be divided into 2 categories [18]

- Geometric-based. The object or the surrounding space is deformed based purely on geometric manipulations. The user manipulates vertices or control points that surround the 3D object to modify the shape of the object.
- Physically-based techniques model the physics involved in the motion and dynamics of interactions. Models simulate the physical behaviour of objects under the effect of external and internal forces.

3.2.1 Geometric based methods

3.2.1.1 Vertex-based method

The vertex-based technique is the simplest of all geometric-based methods. A region of the object in the close vicinity of the collision point is locally deformed. In order to render visual deformation, it translates all the vertices within a certain radius of influence of the collision point, along the direction of the haptic stylus. The deformation function f_d can be determined using a 2^{nd} order polynomial function with no linear deformation($a_1 = 0$).

$$f_d = a_0 + a_2(rd)^2 \tag{3.1}$$

Where, $a_0 = AP$ and $a_2 = -AP/rd^2$. AP is the vector constructed from the coordinates of the stylus tip and the contact point. The radial distance rd corresponds to the distance of each neighbouring vertex, within the radius of influence. The shape of the deformation is related to the degree and coefficients of the polynomial used for the curve deformation [18].

3.2.1.2 Spline-Based model

Instead of directly transforming the vertices of the object, control points are assigned to a group of vertices and are manipulated to achieve smoother deformations (figure 3.3) [18]. This is based on

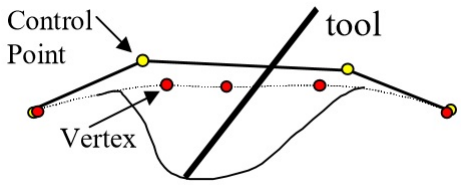


Figure 3.3: Spline based deformation model

the Free-form deformation (FFD) technique [252]. This method is widely used in CAD softwares where users can modify object shapes by repositioning the lattice of control points that surround the 3D object. Any point within the lattice is defined by

$$Q(u,v,w) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} P_{ijk}B_i(u)B_j(v)B_k(w) = BP \tag{3.2}$$

P_{ijk} are the control points, and $B_i(u)B_j(v)B_k(w)$ are known as the third degree Bernstein polynomials or Bezier basis functions. Hsu [120] proposed a method for direct manipulation of free-form surface where points are moved in a way such that the resulting surface smoothly reaches its intended position by means of a least squares solution. Points of a 3D object are moved to a new

location $Q + \Delta Q$. Thus equation3.2 becomes

$$Q + \Delta Q = B(P + \Delta P) \tag{3.3}$$

ΔQ and ΔP are the changes in the position of object point and the control points. Equation 3.3 can reduce to

$$\Delta Q = B\Delta P) \tag{3.4}$$

Change in control points for a given ΔQ can be computed using the pseudo-inverse

$$\Delta P = (B^T B)^{-1} B^T \Delta Q) \tag{3.5}$$

Thus once the changes in the positions of control points are known, the deformed positions of the object can be computed through

$$Q_{new} = B(P + \Delta P) \tag{3.6}$$

This method is efficient to manipulate complex shape, as many CAD softwares already use NURBS to model complex shapes. Thus adding a haptic layer to model force feedback already based on the software layer information used to model the geometry is straightforward. The same control points dedicated to the geometric model can be used.

3.2.2 Particle based deformation model

The particle-based model also called mass-spring model relies on a set of point masses, connected to each other through a network of springs and dampers, moving under the influence of internal and external forces (figure 3.4) [18]. Particle systems have a long history in computer graphics for simulating behaviour of textiles and fluid flow. No explicit equations of motion need to be constructed, which results in a rather simple technique to implement. Each vertex of the 3D object

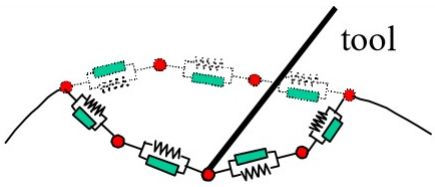


Figure 3.4: Mass-spring of particle method for deformation

possesses a mass and is connected to its neighbours by a spring and damper type interaction. The implementation of such a technique is easy and allows real-time computation. The total force applied at each particle can be decomposed into spring $F_{spring} = \sum k\Delta x$, gravitational $F_{grav} = mg$ and dissipative forces $F_{diss} = bv$ such

$$F_{total} = F_{spring} + F_{grav} + F_{diss} \tag{3.7}$$

Thus, using the Euler integration method, acceleration a, velocity v and position p of each particle can be computed

$$a_t + \Delta t = \frac{F_{total}}{m} \tag{3.8}$$

$$v_t + \Delta t = v_t + \Delta t a_t + \Delta t \tag{3.9}$$

and

$$p_t + \Delta t = p_t + \Delta t a_t + \Delta t \tag{3.10}$$

Viscoelastic models can also be integrated to simulate the elastic damping behaviour by integrating a viscous or damping unit to the mass spring model. The deformation is then function of the force applied and the rate of deformation. It also allows the rendering of phenomenon such as creep.
There are some reported issues when implementing particle method for haptic. To put the system into global equilibrium, a damping term must be added. Increasing this damping causes the system to be stiffer. Similar problems appear when we add multiple constraints. Lastly, an uneven distribution of nodes generates unstable interaction forces and non-smooth graphical deformations. In order to reduce these issues several solutions can be adopted. Stability can be improved by using variables and shorter time steps and considering local deformations. Re-meshing and refinement can help with stability and produces smoother deformations.

3.2.3 Mesh based methods

Numerical mesh based methods relies on the use of mesh to discretise and solve ordinary or partial differential equations (PDE). Figure 3.5 illustrates their general procedure.

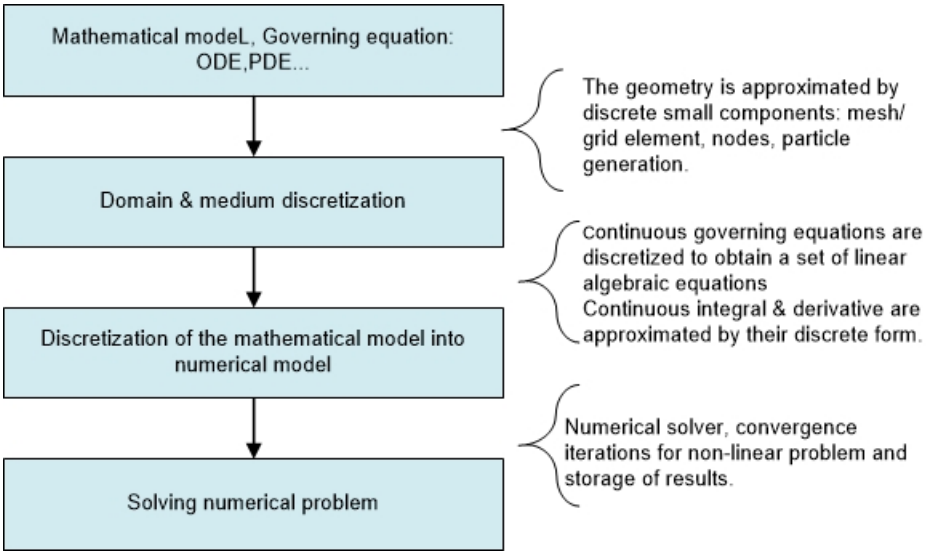


Figure 3.5: Numerical simulations general flowchart

In order to analyse practical problems, domains are split into smaller subdomains made up of geometric primitives like hexahedra and tetrahedra in 3D, and quadrilaterals and triangles in 2D. Discretised governing equations are solved inside each of these portions of the domain. Care must be taken to ensure proper continuity of solution across the common interfaces between two subdomains, so that the approximate solutions inside various portions can be put together to give the field function in the entire domain. Each portion of the domain is known as element or cell and the collection of all elements is known as a mesh.
The numerical solution of PDE requires as a first step the discretization of the system of continuous differential equations into a "discrete" system of simultaneous algebraic equations. Discretization techniques can be categorized in

- Point discretization. This is the framework used by finite difference method, where the derivatives in the PDE system are directly represented at the specified points using Taylor

series expansions of the field function solution of the problem. The collocation method (see section 7.1 p.170) is a point discretization scheme. At locations falling between these specified points, the solution is usually computed using a polynomial interpolation.

- Element discretization. The PDE solution is discretised over small element domains using integral representation. In FEM the field function is represented by a certain function over the cell and then numerically integrated over the volume of the cell. The procedure to obtain the equivalent integral formulation of the PDE system uses variational methods or weight residual method (WRM)(see appendix A p.204).

Nowadays the most popular methods are the FDM, FEM, the Boundary Element Methods (BEM) [17, 24, 43] and Finite Volume Methods (FVM) [80, 128].

3.2.3.1 Structured and unstructured meshes

Structured meshes, also referred as grids, take their name from the fact that a grid is laid out in a regular repeating pattern called a block. The data structure for structured mesh is simple and usually assigns a unique index (i,j,k) to each vertex with coordinate values also uniquely defined for that index-set. Nodes have an equal number of adjacent elements. Structured meshes are characterised by regular connectivity that can be expressed as a two or three dimensional array. This restricts the element choices to quadrilaterals in 2D or hexahedra in 3D. The element topology is fixed and the grid can be shaped to be body fitted through stretching and twisting of the block. Faces and edges of the elements in physical space are mapped to lie parallel to the coordinate axes in the frame in the logical space (figure 3.6). Where non-trivial boundaries are required, "block-structured" techniques can be employed which allows the user to break the domain up into topological blocks (figure 3.6). Algorithms generally involve complex iterative smoothing tech-

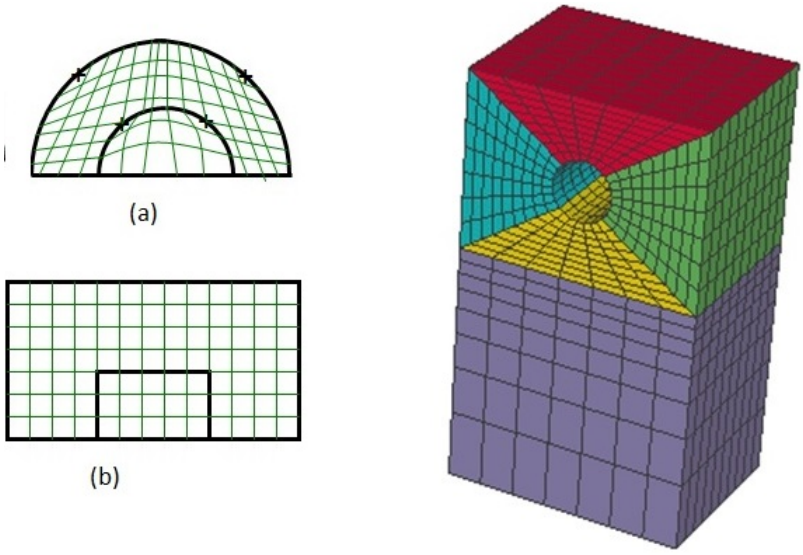


Figure 3.6: Single-block geometry (a) and its logical representation (b) vs multi-block structure mesh

niques that attempt to align elements with boundaries or physical domains. Good structured grid generators utilise sophisticated elliptic equations to automatically optimise the shape of the mesh for orthogonality and uniformity.

Unstructured meshes have been developed to support the use of the FEM. There is no requirement regarding the geometry of the mesh. Such a mesh is usually composed of triangles in 2D and tetrahedra in 3D but can also incorporate any elements of other topologies or a combination of elements of different geometric natures such as trapezoids, bricks or prisms. Elements can be stretched and twisted. Such meshes necessitates the use of a connectivity list to store the neighbouring vertexes of each nodes of the mesh. The storage of such unstructured mesh is characterised by irregular connectivity not readily expressed as a two or three dimensional array in computer memory. Elements possess nodal locations where the interpolation is performed. The spatial coordinates of those vertices are stored into a connectivity list, which indexes those coordinates for each element. The mapping to the global algebraic problem (global vertices numbering) from the local elements (local numbering) requires us to know which elements share a given vertex.

Solving systems of linear equations built over a structured mesh is usually a lot easier. This is because the process of finding neighbouring nodes to build the approximate field at a point of interest is straightforward. The regularity of the connectivity reduces the memory storage since neighbourhood relationships are defined by the storage arrangement.
The regularity imposed by structured meshes makes them inadequate and difficult for the modelling and representation of problems where the domains are complex or possess an irregular shape. Usually the mesh quality deteriorates with increasing complexity of the domain and its internal geometry [106].

Compared to structured meshes, the storage requirements for an unstructured mesh can be substantially larger since the neighbourhood connectivity must be explicitly stored. The data structure is usually more complex than structured mesh and harder to create. Unstructured meshing methods have the ability to be automated to a large degree. Automatic meshing algorithms typically involves meshing the boundary and then either adding elements touching the boundary with an advancing front procedure or adding points in the interior and reconnecting the elements like in Delaunay algorithm. Unstructured mesh generation relaxes the node valence requirement, allowing any number of elements to meet at a single node. The mesh quality is more easily maintain as the domain geometry complexity increase. Complete cover of mesh generation techniques can be found in textbook [85, 212, 213].

3.2.3.2 Finite difference method

FDM has been used to solve approximated solutions of PDE's for boundaries value problems since early 1900 [234, 260]. Cover of regular FDM approximation schemes, convergence and stability results are given in [10, 194, 235, 264, 278, 282]. It is still widely used and appreciated by engineers especially for its simplicity but also its capacity to produce accurate results when the mesh on which it relies on, matches the real geometry of the problem domain. Research on new formulations for fluid flow problem or microwave engineering and optimisation of structured meshes for complex geometry is still a topic of interest for many groups of research. FDM is still used for engineering problems even in micro-scale applications [70, 179, 291, 292, 301, 302].
In FDM, the domain is represented by a structured mesh on which vertices P_{ij} are indexed as shown on figure 3.7. FDM discretization operated on the strong form of the PDE at the discrete points P_{ij}. The differential equation is replaced by an equivalent finite-difference scheme based

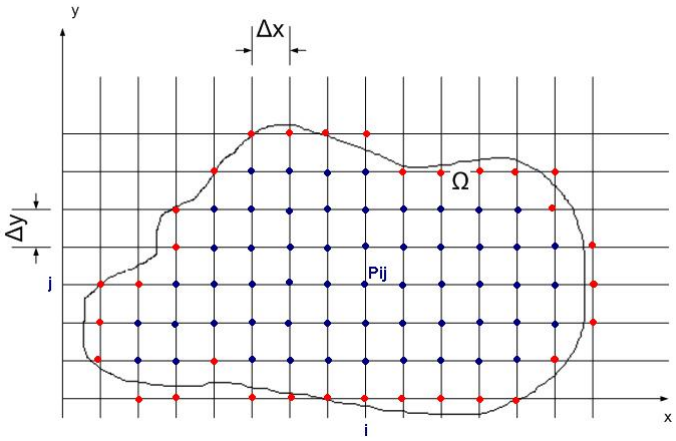


Figure 3.7: Discrete approximation of a 2D region Ω

on the Taylor series representation, whose solution is reduced to simple algebraic manipulations. Famous schemes are forward, backward difference and the central difference. The field function on the studied region Ω is only calculated and known at each mesh vertices.

As shown on figure 3.7, the boundary of the domain Ω is approximated by the red mesh points and there is an error between the position of the real boundary contour and the closest mesh point. As the discrete contour differs from the real boundary $\partial\Omega$, the approximated exact solution T(iΔx,jΔy,m Δt), differs most of the time from the exact one. The more accurate we want the approximation to be, the smaller must be the grid step sizes (Δx, Δy).

When the computations take place over non rectangular domains in two or more dimensions, complying to complex boundary require a very fine step length. Hence, it leads to problems with excessive computation times. This is also needed when the solution varies strongly only over certain parts of the region. In order to avoid an unnecessarily high density of nodes where the field function does not vary significantly, unequal space grid should be used. However non-uniform subdivisions are an issue with the regular FDM formulation.

3.2.3.3 Finite element method

FEM was developed for computations in structural mechanics. It has been a massive area of research in computational mechanics over the last 40 years, to become a major tool in engineering design. Nowadays FEM is considered as the benchmark for its adaptivity, accuracy and ability to address the widest range of engineering problems [25,176,231,249]. It is the first choice for many design engineers because of its ability to tackle designs of complex components and structures. The method possesses a strong mathematical background and robustness [44,62,63,122,125,206, 307].

The behaviour of the system is described over each element in term of the field function values and also the material law(e.g. young modulus, Poisson ratio, thermal capacity etc). The FEM can be seen as a particular type of Galerkin approximation. The Lax-Milgram theorem is the starting point of the method [148].

Using the Galerkin framework, we look for an approximate solution $u_N(x,t) \equiv u(x,t)$ so that the residual function defined by $R(u_N) = r(L(u_N)) \neq 0$ with L being the differential operator of

the parent PDE of the problem. We then have the weight residual written

$$\int_{\Omega} r(L(u_N))W(x)d\Omega = 0 \tag{3.11}$$

The domain of the problem Ω over which the field function $u(x,t)$ is sought, is divided into E discrete finite elements Ω^e such that

$$\Omega = \bigcup_{e=1}^{E} \Omega^e \tag{3.12}$$

Elements depend on the type of problem and also the topology of the domain. The approximation is built over the element itself based only upon the discrete element nodal values [125, 307].

$$u_N(x,t) = u^e(x,t) = \sum_{i=1}^{N^e} \phi_i(x) u_i^e \tag{3.13}$$

Φ_i are the basis or trial function for the element. $u_i^e(t)$[1] represents the unknown nodal values of the field function for the current element. N^e is the number of nodes of the current element e. The basis functions spanning the approximation space S_h are piecewise polynomials as the restriction of each basis function to an element Ω_i is a polynomial [45]. The interpolation should satisfy the compact requirement, which means that the basis function should vanish outside the neighbourhood of their location points.
The order of the finite series approximation is directly related to the type of elements. There is an important variety of element available which are applicable to many different types of problems. Elements are characterised by several features such as

- The shape of the element.
- The order of the interpolation function and the number of nodes.
- The types of nodes (interior or exterior).
- The type of nodal variable, which refers to the field function and

For 2D applications, elements quadrilateral and triangle 3 nodes triangular, 6 or 10 nodes triangular. For 3D applications it is tetrahedron and hexahedron. For many practical problems, engineers use triangular shape elements for 2D problems and tetrahedron for 3D problems. There are the simplest elements, which provide easiness in the assembly process and reduce considerably the computational cost compared to more advances elements.
The finite element method selects a weighting function W(x) such as

$$W(x) = \begin{cases} W^e(x), & x \in \Omega^e \\ 0, & \text{otherwise} \end{cases} \quad \text{for } e = 1, 2, ..., E \tag{3.14}$$

Using this form for the weight function transform the global domain formulation into an element based one

$$\int_{\Omega} r(L(u_N))W(x)d\Omega = \sum_{e=1}^{E} \int_{\Omega^e} r(L(u_N))W^e(x)d\Omega = 0 \tag{3.15}$$

[1]Here an implicit separation of variable is applied. Nodal values are function of time.3.15 as we focus on the spatial integrals

Similarly to the global residual principle, each weighted residual of the elements is forced to zero. This leads to the fundamental formulation of the finite element method.
In the FEM based on the Galerkin method, the weight function W^e is chosen to be similar to the trial function ϕ. Replacing it in equation 3.15 gives

$$\sum_{e=1}^{E} \int_{\Omega^e} r(L(\phi_i(x)u_i^e))\phi_j(x)d\Omega = 0 \quad for \;\; i,j = 1,2,...,N^e \tag{3.16}$$

The next step is the integration over each element subdomain Ω^e. This leads to a matrix system of N^e of algebraic equations. The typical system produced by the FEM is of the form

$$C^e \frac{du^e}{dt} + K^e u^e = b^e \tag{3.17}$$

In the literature, K^e is called the element stiffness matrix, C^e the element capacitance matrix and b^e is the element load vector. Each of these element matrices is assembled into the global system matrix. Then we use regular methods for solving system of algebraic equations such as Gauss elimination, LU decomposition, GMRES, preconditioned conjugate gradient method etc.
The assembly process depends on the type of mesh and element used. It uses a connectivity list for the mapping of each element matrix values within the global system matrix [25,122,231,307]. The domain integral over a sum of element integral (equation 3.17). Element integral evaluation is a crucial procedure in the FEM because the integral of each element must be uniquely evaluated using equations for its side as the limit of integral [122,125,231,307]. The integration is performed in a generic manner on a representative master element. Consider the 2 dimensional case for the sake of simplicity. We assume that a mapping $\mathcal{M}^e$ exists between the element domain Ω^e of real element of a mesh $\mathcal{T}_e$ of the domain Ω, and a so called regular rectangular master element of domain $\hat{\Omega}^e$. In the coordinate system (ξ, η) of $\hat{\Omega}^e$, the sides of the quadrilateral element are fixed and chosen to simplify the integration. Thus one can use regular numerical quadrature technique such a Gauss quadrature, Legendre (see [225] for more on numerical quadrature scheme).

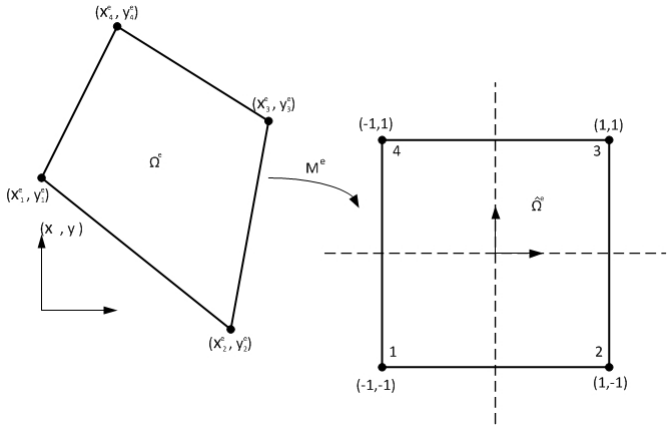


Figure 3.8: Mapping for integration from a 2D quadrilateral (A4NQ) 4 nodes element to its master element

Such mapping makes it possible to evaluate the integrals and derivatives on the simple square master element Ω instead of complex geometry of the original quadrilateral element domain $\hat{\Omega}^e$. A common method or choice of mapping is the use of the same shape functions as the one used for the interpolation of the field function u expressed in the global coordinate system (ξ, η). In that case

the polynomial order for both interpolation is consistent, thus this spatial coordinate interpolation class of elements is called isoparametric [25, 307]. So we have the following mapping

$$\mathcal{M}^e : \begin{cases} x = x(\xi, \eta) \\ y = y(\xi, \eta) \end{cases} \qquad (\mathcal{M}^e)^{-1} : \begin{cases} \xi = \xi(x, y) \\ \eta = \eta(x, y) \end{cases} \tag{3.18}$$

The mapping gives the following expression for the coordinates in the master element

$$x = \sum_{i=1}^{4} \phi(\xi, \eta) x_i^e \quad y = \sum_{i=1}^{4} \phi(\xi, \eta) y_i^e \tag{3.19}$$

Following the discrete element weak form of the PDE, the isoparametric transformation allows us to use a simple Gauss quadrature scheme over the master elements. For instance, considering the 2-dimensional Poisson equation in its weak form (see section B.1 p. 206)

$$\int_{\Gamma_2} v\bar{q} \cdot n d\Gamma + k \int_{\Omega} \nabla v \nabla u d\Omega = \int_{\Omega} v f(x) d\Omega \tag{3.20}$$

Using the FEM procedure we can write

$$\sum_{e=1}^{N} \int_{\Gamma_2^e} v\bar{q} \cdot n d\Gamma^e + \sum_{e=1}^{N} k^e \int_{\Omega^e} \nabla v \nabla u d\Omega^e = \sum_{e=1}^{N} \int_{\Omega^e} v f(x) d\Omega^e \tag{3.21}$$

With N being the number of elements. $d\Omega^e = dxdy$ is the elementary volume. This area is described by the cross product of $|dx \times dy|$. Using the Taylor series equation describing the mapping from $(dx, dy) \rightarrow (d\xi, d\eta)$ and writting the cross product gives

$$|dx \times dy| = \begin{vmatrix} e_1 & e_2 & e_3 \\ \frac{\partial x}{\partial \xi} d\xi & \frac{\partial y}{\partial \xi} d\xi & 0 \\ \frac{\partial x}{\partial \eta} d\eta & \frac{\partial y}{\partial \eta} d\eta & 0 \end{vmatrix} = |J| d\xi d\eta \tag{3.22}$$

Now using the isoparametric quadrilateral element, we map the integration over the parent elements such that

$$\sum_{e=1}^{N} \int_{-1}^{1} v\bar{q} \cdot n |J| d\Gamma + \sum_{e=1}^{N} k^e \iint_{-1}^{+1} \nabla v \nabla u |J| d\xi d\eta = \sum_{e=1}^{N} \iint_{-1}^{+1} v f(x) |J| d\xi d\eta \tag{3.23}$$

Using the Gauss quadrature with n_g Gauss points, which allow the exact integration of a polynomial of degree $2n_g$-1, see [122, 147, 231, 307], we compute the element stiffness matrix for each Gauss point in the element Ω^e. The stiffness element matrix can be written

$$k \iint_{-1}^{+1} \nabla v \nabla u |J| d\xi d\eta = \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} w_{\xi_i} w_{\eta_j} \nabla \Phi_I(\xi_i, \eta_j) \nabla \Phi_J(\xi_i, \eta_j) |J| U \tag{3.24}$$

With w_{ξ_i} and w_{η_j} is the weight associated with the Gauss point ξ_i and η_j. U is the vector containing the (unknown) nodal values at the Gauss point. Once the equivalent set of algebraic equations is solved, the values of the field function values can be retrieved at any point, especially at the vertex of mesh using the interpolation shape functions and the known nodal values at the Gauss points in each element Ω^e. Curve integral is performed similarly for the Neumann boundary condition (the

first term in equation 3.23).

Once the elements of the matrix (equation 3.17) are calculated, each of the term of the matrix is mapped into the global system matrix using a connectivity list (figure 3.9). This connectivity list contains very important data structures, which come with a mesh [85, 125, 307]).

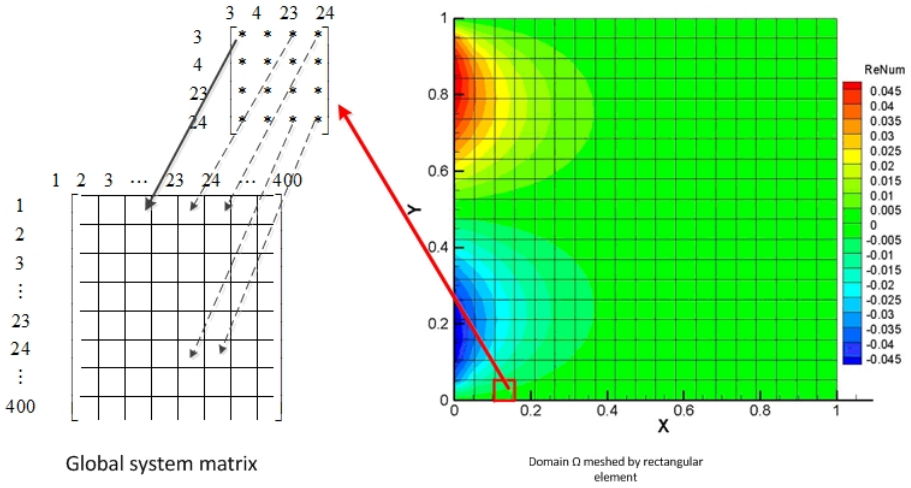


Figure 3.9: General assembly process from element to global system matrix

Similarly, elements vectors are assembled into the right hand side system vector containing the boundary terms. It leads to the final matrix system containing the vector of the unknown field function values at each Gauss point.

Given that the boundary conditions of the problem are satisfied, a unique solution can be obtained for the overall system of linear algebraic equations. The final system matrix is sparsely populated, hence it has to be taken into account for memory storage and solver strategy to optimise computation.

In physical mesh based technique such as FEM, the use of integral formulations provides a more natural treatment of Neumann boundary conditions and discontinuous source terms due to the reduced requirements on the regularity or smoothness of the solution. The FEM is more suited for engineering problems of complex geometries than the FDE because the use of an unstructured mesh provides a more efficient solution to comply with complex geometry domain. High accuracy in region with high gradient can be reached by using a high number of elements.

3.2.4 Overview of methods

Table 3.3 presents a summary of the features of modelling techniques. Geometric-based deformation techniques are faster, and are easier to implement. However they do not simulate the underlying physics. Hence, the emphasis is on visual display and the goal is to make deformations appear smoother to the end-user. Sophisticated physically based models are not suited for fully interactive, real-time simulation of multiple objects in virtual environments due to the limitations in computational power.

Geometric-based deformation techniques are faster, and are relatively easier to implement. But they do not simulate the underlying physics and mechanics of deformations. Hence, the emphasis is on visual display and the goal is to make deformations appear smoother to the end-user. The user needs to define his own interaction model. The force model for instance will depend on the deformation model. A set of linear/nonlinear springs can be considered between the home and

Type	Method	Advantage	Drawbacks
Physics based	Particle based: Mass-spring model Viscoelastic model	Simple. Easy implementation. Real time capabilities. Damping behavior	Needs to be tailored for each application. More Suited for mechanical behavior
	Mesh based Methods: FEM, BEM, FDM	Material properties based. Capability to tackle broad range of physics and Nonlinear problems.	Accuracy depends on Mesh density. Mesh can become a cumbersome.
Geometric-based	vertex-based.	Fast computation	Do not simulate deformation Needs to be tailored for each application.
	Spline based.	Fast computation Smooth variation	

Table 3.3: Overview of the modelling methods

deformed positions of nodes to compute the direction and magnitude of the force vector that will be reflected to the user.

On the other hand, in physically-based modeling, the model automatically computes the magnitude and direction of the field applied to each node. Sophisticated physically based models, although necessary for simulating the dynamics of realistic interactions, are not well suited for fully interactive, real-time simulation of multiple objects in virtual environments due to the current limitations in computational power.

Our goal is to create a tool capable to handle the thermal management of MEMS at the design stage. It should assist the user with his design choices by taking into account the real physics of the MEMS model. Our application will ultimately require the elastic-model for physical deformations and stress model as well as the thermal model. It should also take into account the coupling between the different energy domains for future applications such as electromagnetic or piezoelectric. Hence, it makes the physically based model, the only valid option for our application.

3.3 Implementing mesh based methods in haptic

For haptic applications, the computation load necessary to render the scene in 3D is a challenge. For our application, the thermal properties of the medium must be computed and updated at a rate of 1 KHz to ensure a continuum haptic feedback. Thus the thermal algorithm loop must be computed at the frequency of 1 KHz to provide coherent kinetic feedback [38, 114]. Figure 3.10 represents the flowchart of our typical haptic application.

In all our tests, we are using a Point-object interaction, using the HDAPI routines to manage the CD and extract the data needed for thermal interaction. For the thermal model behaviour implementation, we have used mesh based methods FDM and FEM algorithms. The set of equations is solved by using a Gauss-Seidel or L-U decomposition methods and a regular structured mesh. This is possible because we are dealing with simple geometries in our tests. The algorithms for our benchmark are quite simple but the real time constraints for the haptic rendering makes it a demanding task. Here real-time means that the computation must be fast enough to avoid perception of discontinuity in the haptic feedback (graphic display and force rendering) by the user [20, 114] as described in the previous chapter.

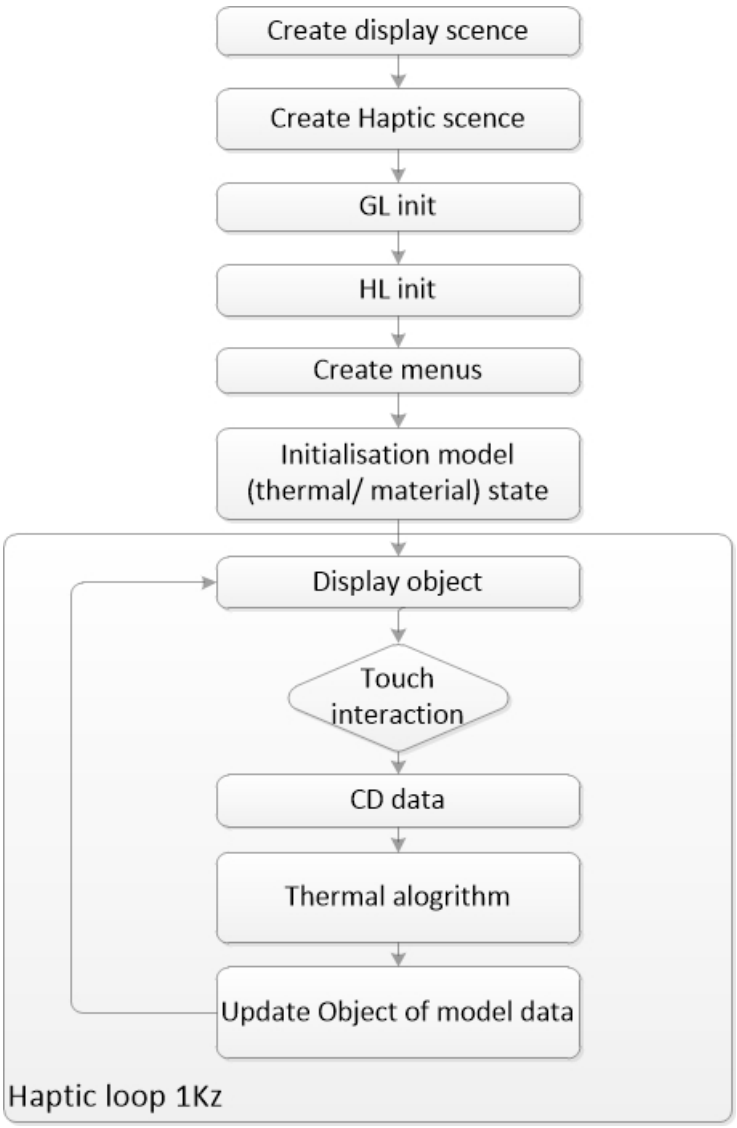


Figure 3.10: Flowchart of the thermal haptic application

There are several strategies that can be used to optimise the implementation of mesh based physics model in haptic application [18].

First multi-threading can be used to optimise the rendering rate. The haptic and visual thread should be separated. The visual thread which manages the display of the scene is running at 30Hz while the haptic loop thread is running at 1 KHz. In order to minimise the memory footprint of the overall model, a shared database is used. However care needs to be taken in accessing the database to limit conflicts. When the haptic loop is trying to write the memory while the graphical loop is reading it to render deformation, this will create conflicts. In order to avoid this problem, synchronisation of the database between the haptic and graphics loop must be ensured. The easiest solution is to use an authorisation flag procedure. When the display is reading the database for graphical rendering at its 30 Hz frequency, the flag is set to 0 and the haptic display cannot access the memory. Once the reading is operated, the flag is set to 1 and the update of the database can be operated.

The second option is to create a hierarchical database for the geometry and its associated properties. This is similar to what we described in section 2.5 page 18. Hence, when operating modification on the database, a copy of the proper hierarchical level can be made in order to operate the

modification. The issue is that the memory space footprint is more important with this strategy. Hierarchical structure of geometric primitives also allows fast access to the local neighbouring primitives of the one involved in the contact point. Thus modifying the local thermal field is easy. The last strategy is to adopt a multi-processing architecture to create a dual database for the multi threading (figure 3.11). One database for the haptic loop and one for the visual thread.

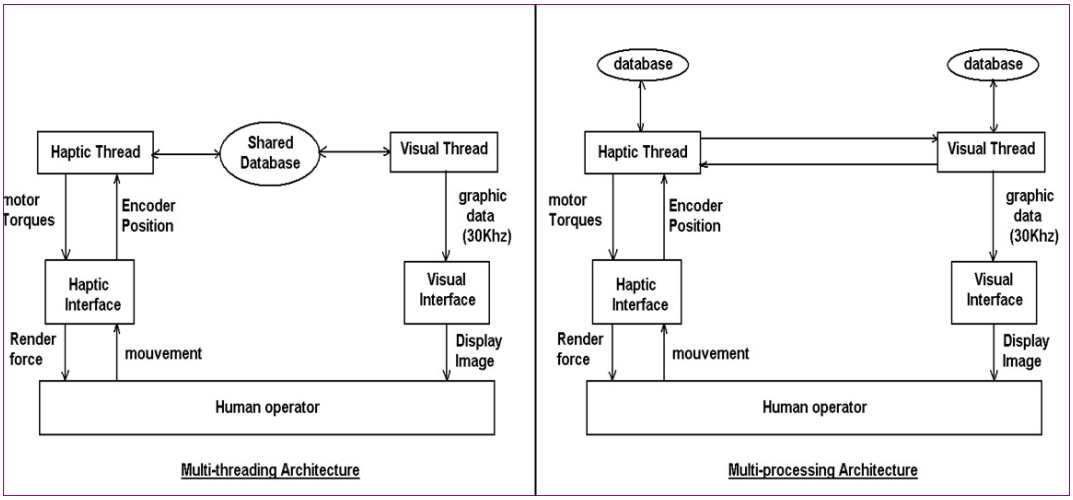


Figure 3.11: Multi-thread vs parallel thread architecture

Both databases are updated at different rates. Synchronisation of both database still needs to be operated and the haptic model database drives the update of the graphics database for the display. When using mesh based methods for the physical based model, the separation of the haptic loop and graphical loop is not sufficient to ensure that the computation of dynamics force, deformation and thermal field at a rate of 1KHz.

A multi-layer architecture can be used to ensure the haptic rate of 1KHz. The haptic loop and display are still run by two different threads. However the physical model is not run in haptic thread which only extrapolates the field or the forces and ensures the rendering at 1KHz (figure 3.12). The model is computed in a separate thread at a lower frequency. Between each physical model cycle, the haptic thread extrapolates the force using a direct model which is much faster. With this solution, the physical model cycle can be modulated depending on the charge of the computation. However depending on the extrapolation technique and the physical phenomenon simulated, a certain frequency threshold still need to be ensured for the physics computation. Otherwise the extrapolation gives an unrealistic feedback to the user.

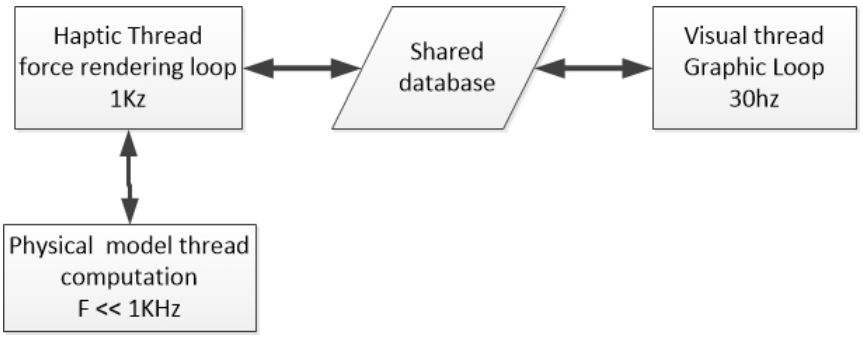


Figure 3.12: Multi-layer threading

Implementing FEM or FDM for haptic model is not an easy task. With FEM, due to the sparsity

of the system matrix the entries of the matrices need to be allocated wisely to save memory. It is difficult to achieve the real-time performance. Moreover, the addition of haptic feedback increases the complexity of the problem. To achieve real-time rendering rates, the inverse of the system matrix K^{-1} can be pre-computed and static condensation (i.e. eliminating unwanted degrees of freedom) technique can be implemented. However, the pre-computation of K^1 is an issue if the topology of object permanently changes during the interaction. For example, if an object is sliced or cut, it has to be re-meshed and the stiffness matrix has to be updated. Mesh refinement is a rather complicated task and can even be unbounded in time. Implementing such paradigm in haptic for real time rendering is another challenge. Lastly if the inverse K^{-1} matrix needs to be calculated using pseudo inverse or preconditioning procedure, it will increase computation time.
The local behaviour of the simulation can be taken into account. For instance, for thermal problems, only the temperature in the vicinity of the contact source can be considered. A radius of influence need to be defined based on parameters such as the type of heat conduction problem and material properties such as heat capacity and conductivity. A Similar approach can be used for deformation where only a zone is considered in the point of contact. With these strategies, we can take advantages of the multi-layer threading. The heat conduction can be computed in the vicinity of the point of contact, extrapolated in the larger area and kept constant for a certain number of haptic cycles for locations further from the point of contact.
As the thermal stress inducing mechanical deformation can be a relatively slow process, a loosely coupling method of force and displacement strategy can be used to simulate the model's body thermal expansion.

3.4 The thermal cue and haptic texturing

A major problem that we had to solve was to determine how to efficiently represent the temperature of the medium. Temperature is a sensation which is interpreted by the second group of the human's somatosensory system sensors [21]. Very few haptic application of thermal feedback have been developed so far [104, 199]. Like the majority of haptic devices available on the market, the Omni device has not been developed to render thermal information cue from the VE. Hence, to enable the designer to "feel" the temperature effect in haptic way, we need to use an artefact to render the temperature distribution of our model.
Haptic interfaces are characterised by the types of feedback provided. In our project we used the Phantom Omni device which provides force feedback through the stylus when its virtual model representation collides with object within the VE. Physical object properties should also be felt with realism. Properties such as physical rigidity, friction, stiffness and roughness can be combined to improve the immersion and realism of experience.
On occasion the colour feedback may be inefficient in presenting the temperature distribution of the simulated system accurately. In some circumstances, the colour is used for others purposes such as representing different kinds of materials or layers. Using colour to differentiate parts constituting the system is common in semiconductor chip and MEMS CAD tools. So the designer should have the choice during the development process for the colour to switch from representing the material layers to the temperature. Another way to represent heat while colour relates to the different kinds of materials is to use an artefact as feedback.
Haptic texturing is the term used to describe the way we can simulate surface roughness or even

surface stiffness [144]. It can enrich the user interaction with a haptic device just as graphical texture enhances visual realism. For our application, it can be used as the artefact to render physical properties like the local temperature of the virtual 3D model. Haptic textures are usually implemented by modulating the surface friction. Another approach is based on the "stick-slip" friction model. The end effector is stuck by using a static friction. If the user applies enough force to overcome this static friction the end effector continue to move until it encounters a new sticking point. Different artefacts have been experimented as haptic texturing solutions.

3.4.1 Stiction effect

The stiction effect is created by generating a force F_s (stiction force) opposite to the normal (**n**) of the surface in contact with the probe (Figure 3.13). The amplitude of the created stiction force

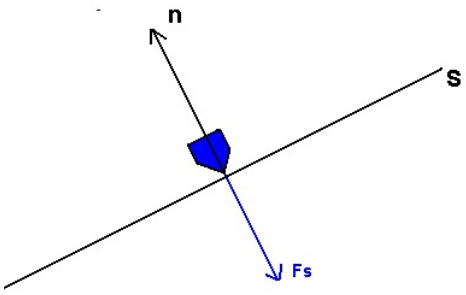


Figure 3.13: Stiction effect force principle

is set by using the temperature at the point of contact or a temperature shading method. This last method is similar to vertex shading and use a weighted average model of temperature based at nodal temperature of polygon in which the area of contact fall. The computation of the normal is generally needed for haptic rendering even without a stiction effect feedback. So we do not need to compute the normal at the surface contact specifically for the stiction effect rendering. Several processes in computer haptic such as rendering a contact force or illumination of a scene in OPENGL requires the computation of the normal at the surface of objects. Thus using this artefact does not add to the overall computational cost. The only problem generated by this method is that the stiction effect can be permanent if the temperature of the body is too high. Sometimes the temperature value may be such that the designer is unable to remove the probe from the surface contact point where it is stuck without difficulty. To avoid this problem, each artefact can be triggered (activated) using the menu of the CAD tool. Friction (see section 2.6.3) is combined with stiction. F_f is coupled to the temperature difference between two vertices and force shading is used to smoothen the stiction resultant force at any point between vertices. Hence, when moving from one point to another on the haptic surface, the user will feel an opposition to his movement which renders the thermal gradient.
In our tests we have implemented and tested the 3 artefacts. Stiction seems to be more natural when designing part whereas friction seems to be better when prospecting for data such as isothermal surface etc. Figure 3.2 shows the graphic interface for selecting the artefact.

3.4.2 Stiffness modification

Another artefact that can be used, is the stiffness modification (figure 3.14). This method consists in linking the stiffness of the material to the value of the temperature at a point of the surface of

contact. When a contact occurs between the probe and the surface of the object simulated; the contact force feedback is modified according to the value of the temperature at this point or in this area. The higher the temperature at this point of contact, the lower is the amplitude of the force feedback generated to create the contact effect. So the user experiences a smoother surface when the body temperature increases.

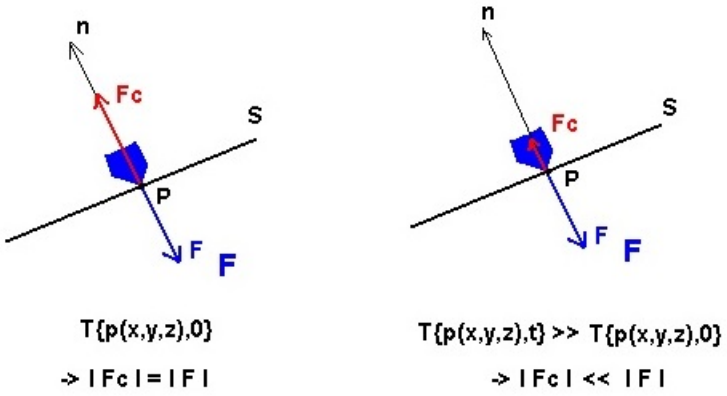


Figure 3.14: Stiffness modification feedback principle scheme

The effect is produced by modulating an opposite force F_c to the normal of the shape of the surface. The stiffness is described by

$$F_c = k\Delta x + dv \tag{3.25}$$

With k being the stiffness coefficient, Δx the penetration depth computed at the local point. d is the damping coefficient and v the velocity.
We have to pay attention when the range of temperature values reached by the surface of the body is relatively too high. In fact, the amplitude of the effect may be such that the human sensors will not be able to understand the meaning of the feedback and estimate the temperature variation. For example, under a certain value of force feedback, it becomes impossible to make the difference between a boundary surface (existence of a surface of contact) and regular movement in the global VE. To partially solve this problem, we have chosen to allow the designer to trigger the appropriate artefact via a menu according to the context of his simulation (range of temperature possible, other effects simulated by the haptic rendering context like gravity, magnetic field etc). Hence the designer can use an artefact effect as the main feedback to locate the higher temperature level area quite easily and switch off this effect when he needs a more accurate view of the temperature distribution. A big problem with this artefact lies in the case of a coupled thermal elasto-dynamic problem. If the system is in movement, it becomes awkward to soften the surface of the model. Beyond a certain point it does not make any sense and the designer just feel like he is handling a soft squishy medium.

3.4.3 Vibration effect

The last artefact that can be used is the vibration effect. A sinusoidal vibration movement is produced by the probe when it encounters a surface. The frequency of the vibration is linked to the value of the body temperature at the contact point. The higher is the temperature, the greater will be the frequency of vibration. This method can be called frequency modulation. The

amplitude modulation, where the vibration force is function of the value of the temperature, has been also tested but it appears to be uncomfortable for the designer after a short period of haptic handling. In practice the force range is really small. Under 0.15 N , vibration effect is elusive and at its maximum level (0.88N) the vibration is such that it is too hard for the designer to work with. Several of our tests show that the more comfortable range of forces is between 0.25-0.30 N. For our tests we have used a vibration force set at 0.26 N and a 5-500 Hz range frequency.

3.5 First haptic implementation and thermal application test

Several tests have been carried out to validate the concept of thermal management using haptic sensing technology.

3.5.1 Haptic thermal feedback tests

The first test we carried out was to assess the efficiency of haptic texturing for thermal feedback. The first method that we used to represent the temperature distribution was colour intensity. When no other information is required by the designer on the physics of the model, this simple method is efficient and allows a fast understanding of the temperature gradient. The main reason is that the sensitivity of the rods of our photoreceptors in the retina are much important than the colour (spectral) detected by cones. For the designer, it is more natural to have a temperature feedback with only one colour. The darker the colour of the body, the lower the temperature value.
For this test, we consider a simple Laplacian problem on a 2D surface, like a substrate with only Dirichlet boundary condition. 3 edges are insulated and one is kept at constant temperature T_a . We use FDM and FEM using QUAD element for the temperature distribution developed in C++ and integrated as physical model with the HLAPI. Figure 3.15 shows color intensity representation for temperature.

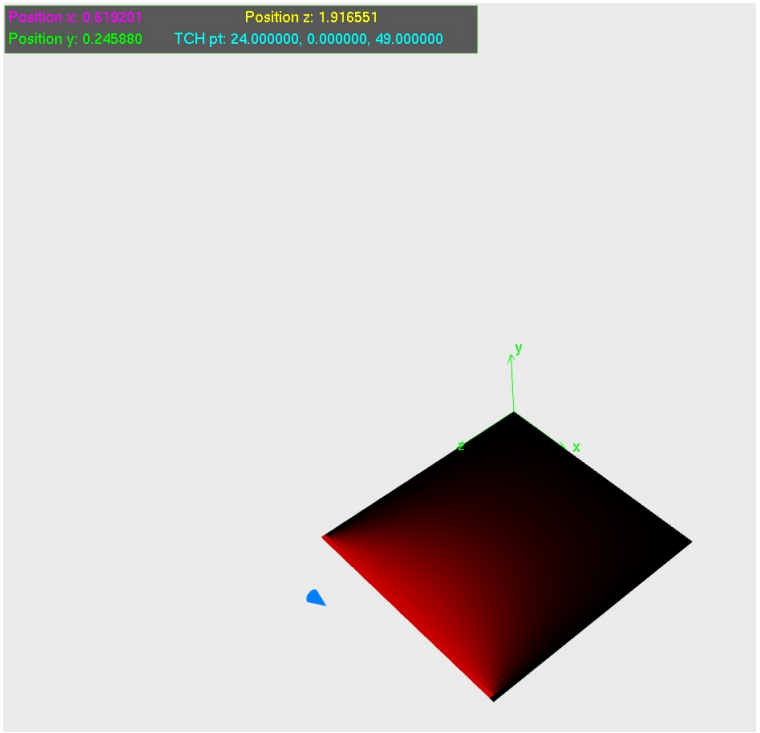


Figure 3.15: Haptic CAD software test: haptic thermal rendering for steady state of a 2D plate

Along with the colour representation we coupled haptic texturing and made perception tests with a group of student and researchers to analyse their perception of thermal parameters. The 3 methods of haptic texturing (see section 3.4) were tested e.g. stiction/friction effect, stiffness modification, and vibration. The first part of the test was made without the use of a colour gradient display. Only the haptic texturing techniques were used to render the temperature field. For all the techniques, users were able to assess the variation of the effect along the surface model. The greatest perception of temperature intensity was assessed through the use of stiffness modification and vibration. But the concept of gradient seems to be better grasped using the stiction model. Table 3.4 summarises the results for the different method of haptic texturing.

		Color **gradient** **(CGr)**	**Stiction**	**+ CGr**	**Vibration**	**+ CGr**	**Stiffness**	**mod** **+ CGr**	**Color**
Perception of temperature intensity	2D	G	G	VG	G	G	VG	ok	VG
	3D	G	G	VG	G	G	P	Conf	VG
Perception of temperature gradient	2D	Opt	G	VG	AV	G	AV	G	G
	3D	Opt	G	VG	G	G	Bad	Conf	G
Temperature measure		P	VP	P	Vp	Par	VP	VP	Opt
Perception of shapes		AV	VG	VG	Para	Par	bad	bad	AV
Comfort		G	G	G	No	Unconf			G
Natural feeling		VG	G	Opt	No	Conf	Conf	Cont	G

G=Good; VG=very good; Av=average; VP=very poor;
P=Poor; Opt=Optimal; Par=parasited; Conf=Comfortable

Table 3.4: Summary of haptic texturing results for thermal haptic feedback tests

When blind tested and without prior knowledge of the physical effect rendered, no user was able to guess that the physical phenomenon rendered was the temperature field. Adding the colour gradient channel clearly increases the awareness regarding the type of phenomenon experienced. Most of the feedback from user was that the haptic rendering is clearer when coupled with colour representation. In this configuration most users report that stiction/friction method is the most comfortable and natural feedback and that the effect seems to be more coupled with the colour representation. Vibration was mostly reported as awkward and stiffness was perceived as the material property such as rubber or steel.

To further investigate on the relevance of haptic texturing for thermal rendering, another blind test was conducted to measure the impact on the perception of shapes. The shape of the model (plate, ball, and cube and slab in 3D) was made transparent in the graphics rendering pipeline. For this test we imposed an analytical temperature gradient through the different model. The result is that objects could only be felt through the haptic device interaction. Only the center of the

object was represented in the VE as a hint of the object position. This allows the user to find the object with the stylus and initiate the interaction in the VE. The results confirmed the ones from the previous test. The combination of stiction and friction still allowed the user to differentiate the shape of the different models. The difference between the slab and the plate could be noticed and temperature gradient information was still perceived by the users. Vibration still allowed the perception of shapes but only with moderate intensity. However the perception of temperature gradient disappeared for almost all the users. The result for stiffness were even worse than in the previous case scenario. Stiffness modification on 3D shapes is perceived by most users as the type of material that the model is being made of.
Following those tests, stiction/friction effect appears to be the best candidate to render thermal feedback. Another issue with vibration is that it could interfere with the movement of deformable object during their manipulation. For instance, if investigating the response of sensor, adding vibration to the force rendered locally by the haptic feedback could be counterproductive in term of perception of the sensor beam's displacement.
The case of negative temperature was not considered for these tests but should ultimately be considered for the purpose of creating a haptic thermal management tool for MEMS design. These rendering effects should also be tested with deformation of the elastic model in order to assess how the dynamic deformation of the model affects the perception of the haptic thermal cue.

3.5.2 Haptic thermal exploration tool

An issue with the haptic texturing method for thermal feedback is the relative incapacity of the user to define the temperature range or absolute value. For that purpose colour scale representation is more accurate and straightforward for the designer as he only needs to refer to the colour bar legend in order to find out the temperature range value associated with the local colour. In all our previous tests, users were unable to assess the local temperature value based on the haptic texturing feedback alone. The problem is similar with one colour representation method. Intensity of the color or haptic texturing effect only allowed the user to know that the temperature is locally relatively hotter or cooler.
To resolve this issue, we implemented a paradigm we called haptic thermal exploration tool. This idea is to choose a threshold temperature. Below this temperature, the local haptic texturing is very mild. If the local temperature is over the threshold, the effect becomes much more intense. This way, the user can track isothermal line through the model using the haptic texturing feedback. The feedback can be set for a particular range of temperature, thus facilitating the exploration.
To experiment this principle, simulation of the temperature through a 3D slab representing a substrate plate has been conducted. The initial temperature of the 3D slab is assumed to be at constant ambient temperature T_a. All faces of the 3D cube are assumed to be insulated except one at fixed temperature.
On the figure 3.16, we are only able to see the temperature at the surface of the plate. For 3D design and data exploration this representation is incongruous and may lead to a wrong appreciation of the field. In order to solve this issue, one can use regular clipping plan. This method is well known in mechanics for part assembly. Different section views are drawn to provide better understanding of the shape and the kinematic of the system. It consists in clipping the body to gain access through a side view to information that it would be impossible to see with a normal display. In the case of our 3D plate, it is relevant to investigate the temperature distribution inside the slab.

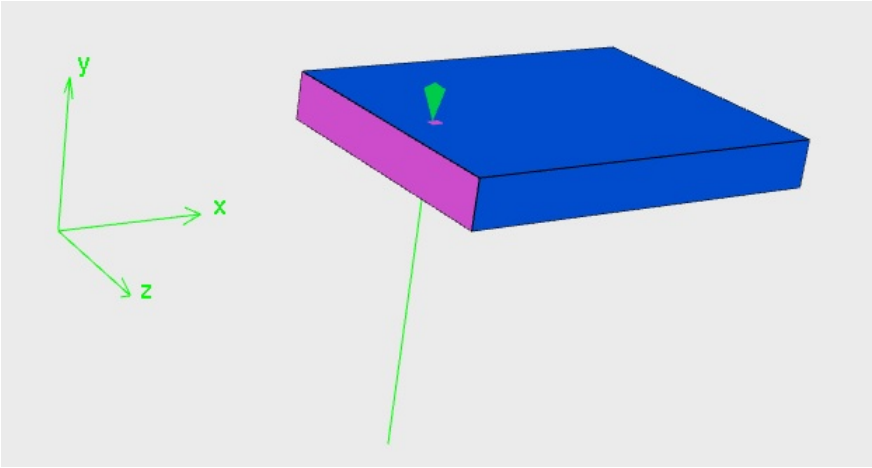


Figure 3.16: Simulation of a steady state temperature distribution through a 3D slab

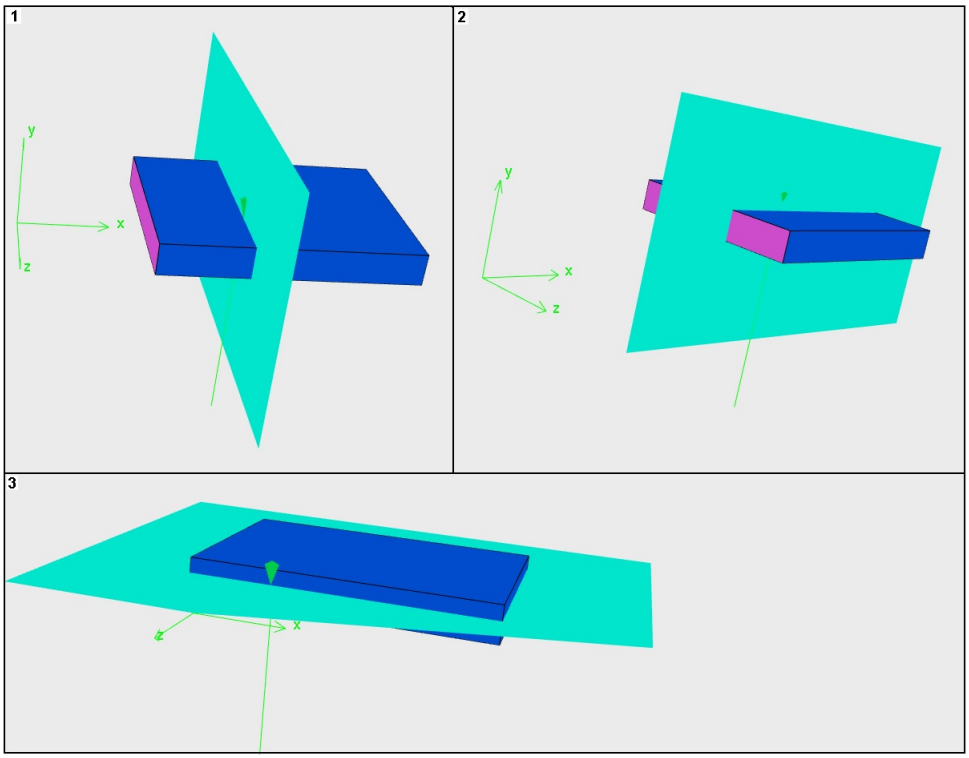


Figure 3.17: Clipping plan according each frame axis

The process is relatively simple. At any time, when the designer receives important stiction feedback; he can choose to activate the cutting process through the menu. Hence, he can choose the clipping plan (figure 3.17) and which part of the body is going to be made transparent. The designer can remove the upper or the under part regarding to the plan. After setting these options, the designer can adjust the clipping plan along the chosen direction and cut the shape along this plan. Figure 3.18 shows the result of the operation for several cuts. Once the model is cut the designer can use the probe to track particular gradient line which is made easy through the stiction artefact. Further cuts can then be made to isolate a particular part of the model.

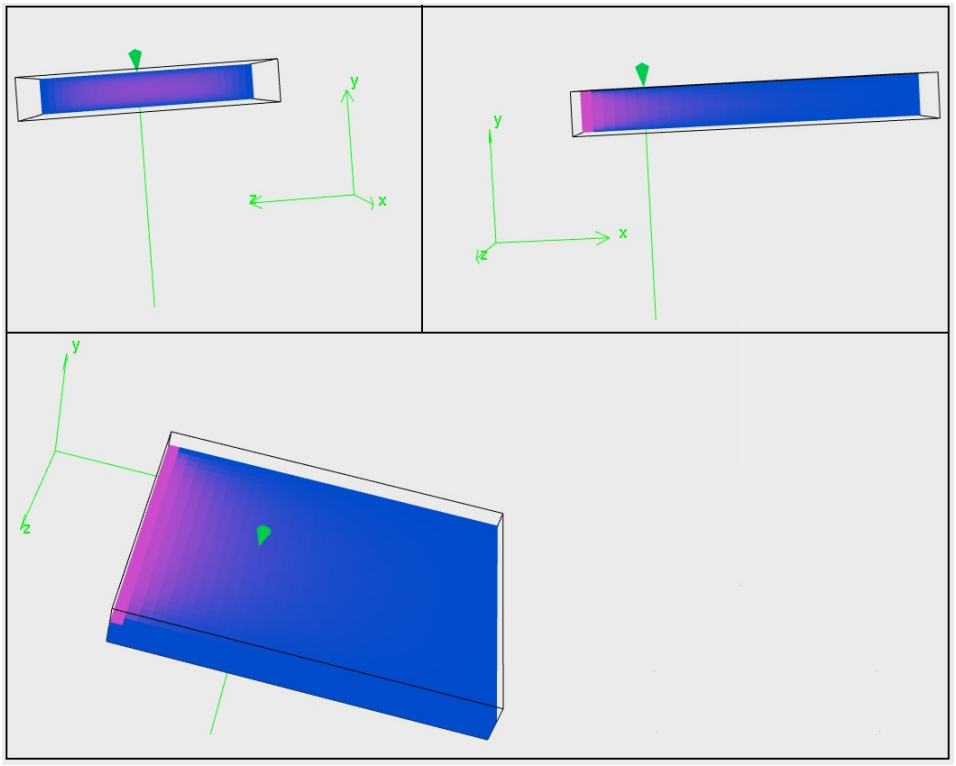


Figure 3.18: Different section views after applying the cutting process along each plan

3.5.3 Isothermal surfaces and cutting issues

Cutting method is an interesting tool to investigate the temperature distribution through the design. By selecting a temperature value point using the probe, the designer could also be interested in drawing the isothermal surface at the same temperature value. This method provides some useful information for designers. It can allow us to verify if a certain temperature value is reached within a specific critical part of the body. For instance, the designer using the haptic probe as a heat source may apply it on the surface of the model. Using this isothermal representation, he can move the source and observe in real time the variation of isothermal surface. This could be an advantage when designing the different thermal zones of MEMS.
One may be interested by cutting the body along its own custom surface. By pressing the second button of the probe, the designer activates the cutting option. Hence he has to draw a rough contour by which the software used to build a surface. The next step is to cut the body along this surface and remove the part which is under or upper the surface. This method has not been implemented yet in our tool but a simple form is tested. It consists in drawing a plan surface built on a selection of 3 points selected by the designer. The temperature is plotted on the surface. With this type of application, we encountered the first problem in using mesh based methods. These issues are similar in a sense to those encountered in haptic surgical simulation with cutting process. Cutting the domain along isothermal surface would require

- Mesh refinement in order to evaluate the temperature field along the surface.
- Elements conform to the isothermal surface plan defined by the cut.

Processing such operation in real-time is definitely not an easy task as dense mesh is requirement for fine approximation of the field on the desired surface. Hence, it seriously brings issue in respecting the frequency update of the model for real time computation.

Contrary to standard modelling approaches concerned with fixed mesh topology and which are used in interactions between virtual instrument and object, cutting modifies the topology of the model significantly. Cutting problems are well known for their complexity in algorithm. The process is akin to a localized adaptive refinement of the simulation mesh. There are many pitfalls that come with adaptive simulations hence with cutting process simulation. Some of the issues that are encountered are

- distorted or ill-shaped elements,
- drastic increase in the numbers of elements after multiple cuts process at the same location,
- incompatible elements in the cut region.

Hence it hampers the solving of the underlying mechanical model.
Most the previous implementations either rely on FEM [46, 66, 182, 250] or mass spring model [52, 73]. At the time when the issue was encountered in the work of this thesis, FEM solution for such problems were highly reliant on the introduction of considerable simplifications for real-time applications [193,204,218]. Methods such as condensation [46] and adaptive simulation [72] were among the proposed strategies to obtain real-time performance. Reduction of the mesh to only surface elements [130] have also been investigated. It maps the 3D problem into a 2D auxiliary surface to simplify the FEM calculation.
Mass-spring models are widely used in simulating the cutting of soft tissue since it is relatively simple and easy to implement [37,47,67,216]. Particle methods have also been investigated [129]. There is two type of model:

- Volumetric models such as tetrahedral models are chosen since they can simulate objects with an interior structure. However, topology modification of volumetric models is extremely complex. For instance for tetrahedral elements cut by planar surfaces it depends on the number of cut edges and intersected faces [37], leading to different topology scenario.
- Surface mesh models are relatively easy to manipulate compared to volumetric models. However normal surface models cannot denote object interior structure for cutting results.

Progressive cutting modelling techniques can be categorized in two categories. The first one considers that cuts should occur as the user moves the cutting instrument through the object. In the second model, the user follows a specific cutting path by performing several cuts and joins them together. The model should then join together those cuts if close enough.
Implementation, of such realistic simulation of cutting is still an open area. Recent achievement have been carried out using FEM model with efficient topological operation using GPU parallel computing [68, 215].

3.5.4 Haptic management and thermal expansion

The test consisted in simulating a transient heat conduction problem. Basically it was the same problem as the previous one but introducing the conductivity value of the substrate plate, here aluminium nitride (AlN). Usually, the value attributed for the conductivity λ of a substrate AlN is around 170 W/mK. For pragmatic reasons, we have carried out our tests with conductivity values that enable us to observe the transient phenomena. Figure 3.19 shows some screenshots of the heat wave propagation after applying the Boundary conditions with the haptic probe.

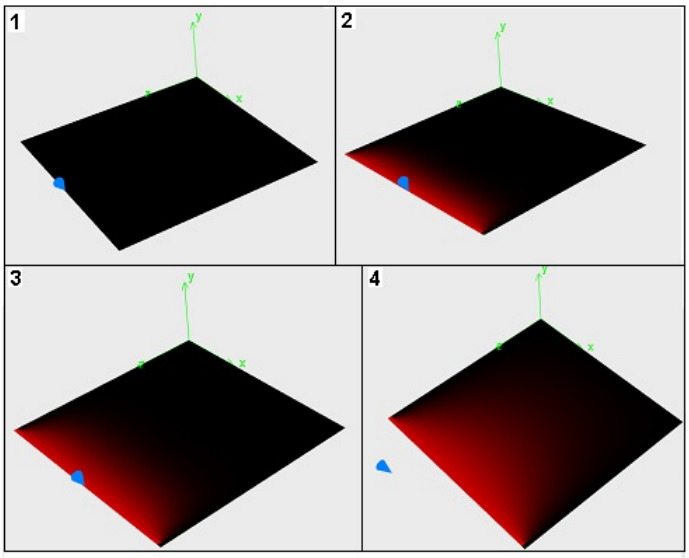


Figure 3.19: Haptic CAD software test: 2D plate unsteady state

Another test was to render the temperature expansion through a 2D substrate plate and with local heating source applied by the probe of the haptic device. The aim is to simulate in real time temperature expansion that could be applied to thermal management of MEMS for packaging design. For instance, how soldering or other external heat sources can interfere with the encapsulated sensor. Thermal conduction model in real-time is needed for this application. One of the major concerns for designers is to be able to apply heating sources in real-time directly by using the haptic device. For simplification we assumed that the source is a punctual one. Figure 3.20 shows the propagation of heat through the plate while moving the heat source around with the haptic probe.

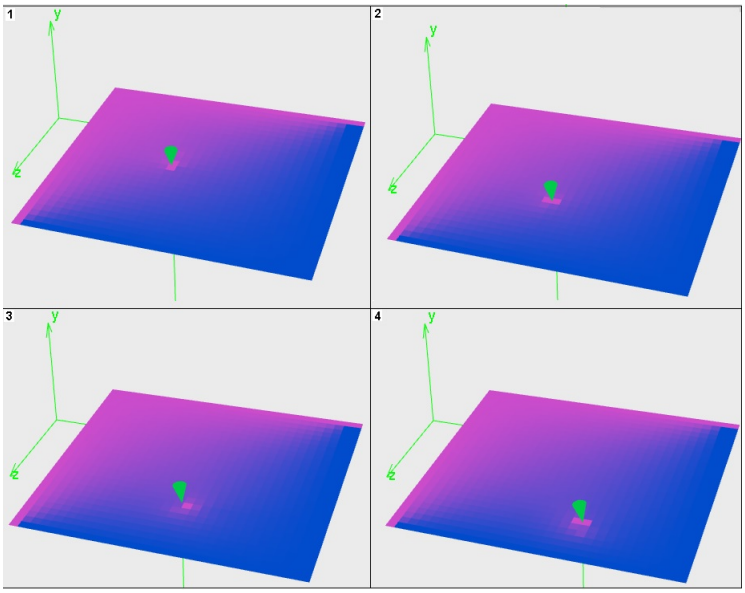


Figure 3.20: Heat propagation from point heat source on haptic 2D plate

The designer can use the probe as a heat source to apply a temperature source locally and observe in real-time the modification of the steady state temperature distribution. FDM model was implemented for this test. In these tests the mesh model shows its limitation with regards to computational speed. In order to ensure a smooth haptic texturing feedback, sufficient mesh density is required for the mesh. This is especially the case for 3D model.

This issue has already appeared for benchmark problem with simple shapes. Thus it appears rather difficult to foresee implementing a FEM model for a complex MEMS structure with multiple layers, along with a mechanical model for deformation on top of the thermal one. The computation charge is too much for regular architecture and computers.

3.6 Mesh based method issues

The rising complexities in terms of domain representation, multi-scale and crossed discipline of practical problems have pushed the industry to choose FEM as their main simulation tool. Its robust mathematical foundation provides for risk assessments and cost saving analysis, a very useful tool which has found its place in many PLM platforms. The meshing process for 3D problems is complex and a computationally expensive step in the process of solving practical problem governed by PDE.
The meshing process requires insight from engineers to decide on a sensible accuracy during the analysis stage. If the analysis process finds fault with the design (figure 3.21), the process will go back at the CAD environment stage.

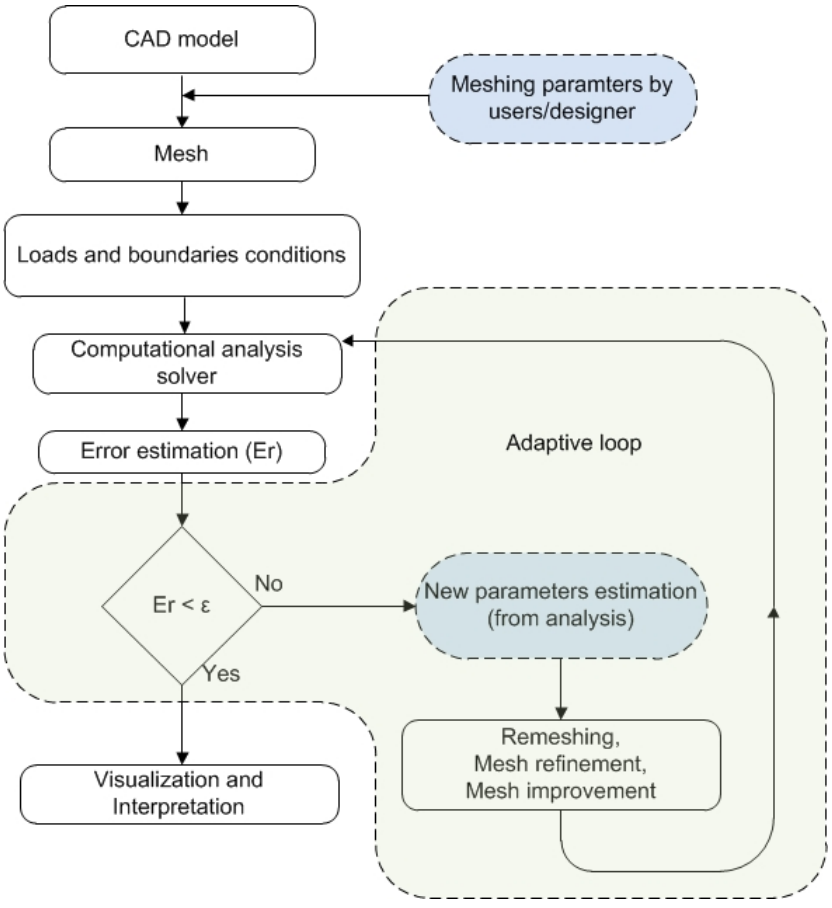


Figure 3.21: Typical adaptive simulation process with re-meshing flowchart

Meshes need to respect the quality criteria. For instance the mesh elements should not be squeezed such that the associated Jacobian for the mapping would be degenerated. At some point, with large deformations, the mesh distorts in such a way that it impacts the accuracy. The worst case scenario includes convergence problem or simulation break down. The solution is usually provided with the automatic re-meshing adaptive process in the adaptive FEM framework. These powerful

techniques are however very demanding in term of computational power, time and experience. For 3-dimensional problems, they call for very complex parallel implementation with management of data through efficient load balancing techniques [82, 272, 285].
The re-meshing process also requires us to interpolate unknown field functions e.g. displacement or stress, or in our case the temperature gradient corresponding to the current solution, which intrinsically can introduce errors of diffusion type. Special FEM has been developed to overcome this dependency towards the mesh in certain cases by the use of enrichment techniques. These techniques are somewhat not general in their applicability and would require insight and prior knowledge about the problem solution.
In the view of haptic design, the dynamic shape modification procedure, whether with direct interaction of the user or not, is very complex as the domain has to be re-meshed and updated. The real-time interaction constraint makes this task computationally expensive for FEM mesh based model without reduction or simplification.

3.7 Conclusion

In this chapter, we have presented an overview of the different challenges inherent to the development of a haptic tool for the thermal management of MEMS design. Following our implementation and test, the first results in using haptic for thermal information feedback are promising. However, the update rate requirement of the haptic architecture, CD technique and type of model for object in VE are very important concerns. The complexity of the model limits the performance of the CD detection. A trade-off between the model's complexity and the computation speed has to be made in order to meet the criteria to ensure the haptic's real time feature.
Using FDM for the computation of the thermal model has enabled us to develop a first version of our haptic tool application. This allowed us to validate the overall concept of haptic rendering for thermal management. Unfortunately, this method soon showed its limits in term of flexibility. The constraint on the density of the regular grind used to discretise the model severely hampers the accuracy of the method. Another issue is that FDM leads to a coarse approximation of the temperature distribution in the neighbourhood of complex contours due to the use of regular structured grind. FEM was another considered solution but the implementation of a demonstrator using FEM brought to light major issues with regard to meeting the update frequency for real time rendering. The issues in practical implementations have led us in the direction to prospect for new numerical schemes, which would better suit the haptic framework and make the most of its potential for future development. The computational cost associated with methods relying on mesh has pushed for research into solutions with the same advantages as FEM, but without mesh. This has led us to the direction of meshfree methods.
Meshfree methods have been developed since the end of the 70^s in order to limit the dependency of the approximated solution on the underlying mesh of the problem domain. The idea was to solve PDEs only using arbitrary constellation of nodes in order to overcome the problems embedded by the use of meshes. Large deformations, fractures and moving boundaries problems are types of problems where meshfree have attracted a lot of interest due to their formulation and the absence connectivity and need of re-meshing.

Chapter 4 Meshfree methods

The main goal of Meshfree Methods (MMs) is to eliminate the mesh structure required by conventional solvers by constructing an approximation of the field variables based entirely on a random constellation of nodes. These methods are often referred to as meshfree, meshless or free-grid method in the literature. A meshless method is an algorithm in which the definition of the shape functions depends only on the node positions. The evaluation of the node connectivity is bounded in time, and should only depend on the total number of nodes in the domain. MMs have experienced tremendous development in the past 20 years.
The development of the smooth particle Hydrodynamics (SPH) in the 70's and its first implementation in mechanical and fracture problems gave rise to strong interest for these methods, thereby leading to further research to correct and overcome their original limitations. In the pace of the RPKM and EFG methods developed originally to overcome the lack of accuracy and consistency, sensitivity of particle distribution and also stability, numerous meshfree methods have been developed.

4.1 From particle method to mesh-free Approaches overview

MMs can be globally break down into two different categories i.e. particle methods and meshless methods. The Meshfree Particle Method (MPM) introduced the idea of building approximation with no underlying mesh [187]. They were originally pure Lagrangian methods ,developed for astrophysical applications such as binary stars and stellar collisions, collapse and formation of galaxies and even simulation of the evolution of the universe.
The continuum at a certain time is discretised by a set of particles moving in the Lagrangian frame according internal interactions and external forces. The size of particles varies from nano-scale to macro or astronomical scales. Each particle is either associated to one physical object or a part of the continuum modelled domain. Each particle is also associated with a set of field variables (mass density, energy, velocity, position and momentum). The evolution of the variables of the system is governed by the fundamental law of conservation of mass, momentum and energy. The material properties are able to move under external forces and also internal interactions. The PDE is transformed into equations of motion for the discrete set of particles such that the particles can be moved via these equations. After time discretisation of the equations of motion we obtain a certain particle distribution for every time step. Therefore, we get an approximate solution to the PDE via the definition of a density function for these particle distributions. These methods are easy to implement. However, they exhibit relatively poor convergence properties in weak norms.
In MPM the fields functions and derivative are approximate at a particle location using only the information of the neighboring particles within a so-called influence domain. The approximated field function u at the particle location x is written as a weighted average of the neighboring values

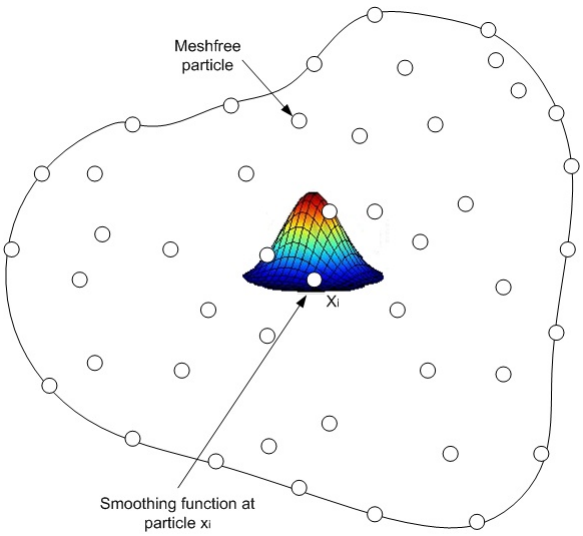


Figure 4.1: Smoothing function at a particle at the location x in its support domain

$u_i$of the N surrounding particles.

$$u^h = \sum_{i=1}^{N} \phi_i(x) u_i \tag{4.1}$$

u_i and ϕ_i are respectively the field function value and the smoothing function at the i^{th} particle.
One of the widely used MPM is the Smoothed Particle Hydrodynamics method(SPH). The discretisation based only on particles with no fixed connectivity and the Lagrangian formulation is very well suited for large deformations and multi-scale problems. The field variable approximation at each time step is updated with the current surrounding particle distribution, which provides an effective way to handle large deformations. The inherent adaptivity provided by the particles can also allow complex geometries to be handled simply. In the mass-packet method [300], mass is subdivided into small mass packets of finite extensions. Each is equipped with many finitely internal degrees of freedom. These mass packets move under the influence of internal, external forces, the laws of thermodynamics and can undergo arbitrary linear deformations. This leads to the development of the relatively new finite mass method [88, 151].
Meshless methods also referred as meshfree (MF) methods follow a different approach. They are based on the building of approximations of the field function only on a set of scattered field nodes (figure 4.3). The first important part of meshfree methods is the discretization scheme. Meshfree shape functions can be used with weak formulation(global Galerkin or local), collocation, or a mix discretization process to set up a linear system of equations. Figure 4.2 shows a comparison of the general procedure for these meshfree methods with FEM.
The first challenge faced in meshfree methods is in building a robust approximation based only on a constellation of nodes, without pre-defined elements as in the classical techniques of FEM or FDM. The shape functions are built upon local support domains (figure 4.3) which are patches or volumes attached to each nodes whose union forms an open covering of the domain.
Local shape functions are then constructed using data fitting methods. This step is probably one of the most important and critical issues in meshfree methods as they play a key role in the implementation and accuracy of the method.

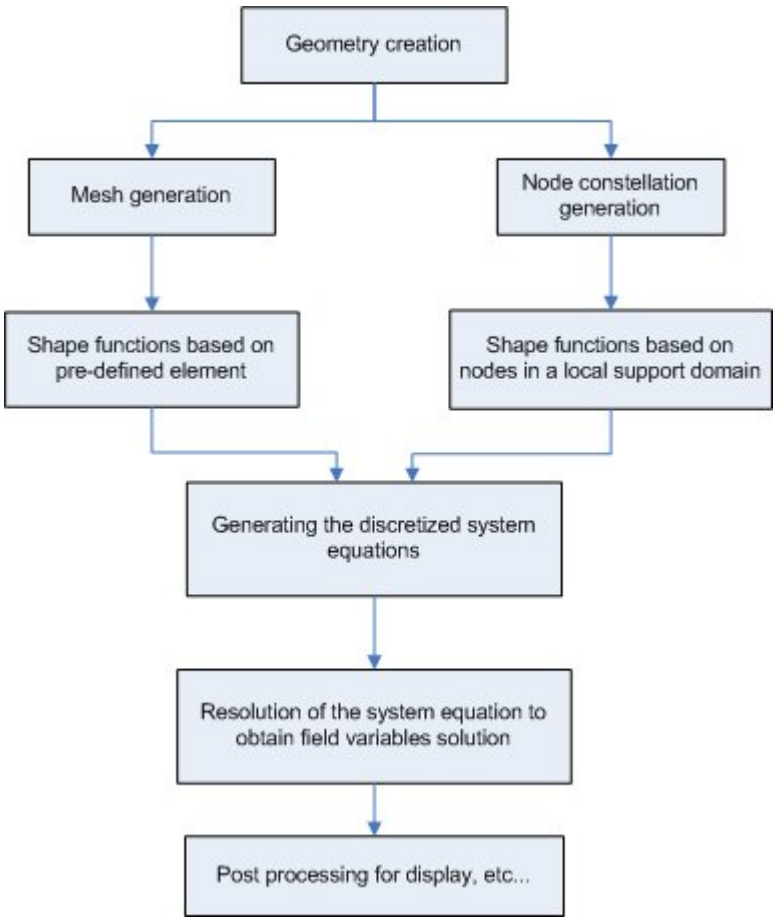


Figure 4.2: Meshfree versus FEM general flowchart

4.2 Meshfree shape function

Meshfree shape functions (SFs) are built using interpolation methods on a set of scattered nodes. Given a set of sample values at certain locations, the idea is to find an approximation u^h which exactly match the given measurements at the corresponding locations. Meshfree approximation schemes to build shape functions fall into three main approaches

- Integral representation like SPH [30] and the reproducing kernel method (RPKM) [8].
- Finite series interpolation like the moving least square (MLS) scheme and radial point interpolation method (RPIM) using radial basis functions. This is also used by Babuska and Melenk with their partition of unity method (PUM) [15, 184].
- Finite differential representation methods which fall under the finite point method [173,208–210]. The idea was derived from the FDM but for irregular nodal distribution.

4.2.1 Smoothed Particle Hydrodynamics method

The SPH is considered the starting point of meshfree methods. Lucy [174] and Gingold [92, 93] were the first to employ the method to solve astrophysical problems in two and three dimensions. In the study of stars and galaxies evolution, the collective movement of particles is similar to the movement of a liquid or gas flow and can be modeled by the governing equation of classical Newtonian hydrodynamics. Liska proposed a similar method that was inspired by the finite difference

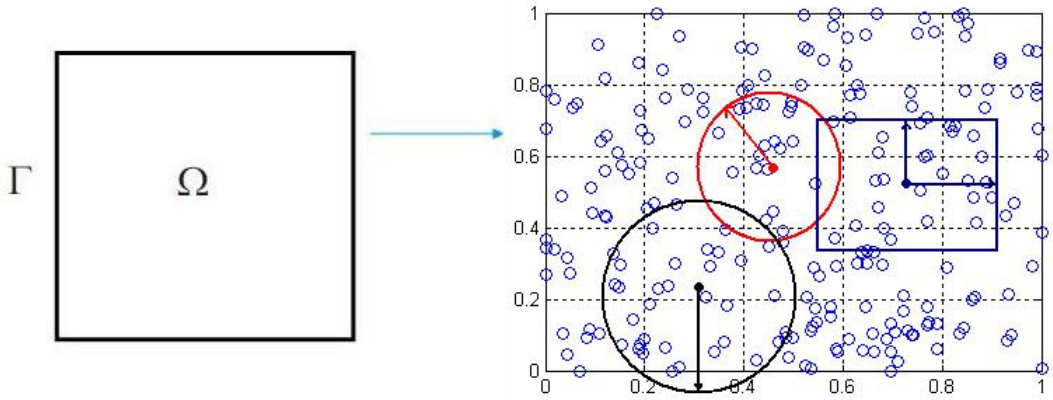


Figure 4.3: Nodal arrangement for domain Ω and local support domain

formulation [156] and later developed much more effectively by J.Monaghan [187, 190].
Until 1985, the SPH method was dedicated to astrophysics applications. A first review of the methods was published by W.Benz [33, 34] and Monaghan [187] who were pioneers in applying the method for modelling of impacts at high speeds with penetration or damages [188, 192]. SPH was adapted to solid dynamics in 1990 [154]. Libersky [153] extended the method to work with full stress tensors for 2D and 3D applications where the material strength was crucial in simulating the dynamic material response. Further studies revealed the problems inherent to the method [28, 113, 273, 274] identified the following shortcomings

- Lack of consistency,
- Tensile instability [16, 77, 191],
- Zero energy mode [40, 125, 227, 288, 305],
- Boundary condition enforcement [54, 153, 229].

A comprehensive review of the shortcomings and developments to overcome them can be found in [287]. The research on trying to solve these shortcomings led to the development of many other meshfree methods. An example is the moving least squares approximation used in EFG [280] that was originally proposed as a completeness correction by Belytschko. The method has since been applied to a wide set of applications ranging from mechanics, conduction [64] to the analysis and simulation of brain contusions [105]. So far, the SPH method has demonstrated a tremendous capability to treat engineering problems with extremely large deformations such as explosions, impacts and penetrations [188, 270]. This method has been applied for modelling the universe [189], the formation and evolution of galaxies [36, 217], or in the metal forming process. An excellent review of recent developments can be found in [165, 237, 287].

4.2.2 Reproducing Kernel Particle method(RPKM) or corrected SPH

In the RPKM, in order to enforce a certain degree of consistency or completeness, a corrective function C is included in the finite integral formulation and multiplied with the original kernel. This idea was originally developed by W.KLiu [167–170] and is the key for the reproducing kernel particle method. The approximated field function becomes

$$u^{\rho} = \int_{\Omega} C(x, x-\zeta) W(x-\zeta, \rho) u(\zeta) d\zeta \tag{4.2}$$

We can rewrite the approximated field function with the conventional form

$$u^{\rho} = \int_{\Omega} \tilde{W}(x-\zeta,\rho)u(\zeta)d\zeta \tag{4.3}$$

with

$$\tilde{W}(x-\zeta,\rho) = C(x,x-\zeta)W(x-\zeta,\rho) \tag{4.4}$$

$C(x,x-\zeta)$ is the correction function,also called corrected reproducing kernel function and is expressed as a linear combination of polynomial basis functions is of the following form

$$C(x,x-\zeta) = c_0(x) + c_1(x)(\frac{x-\zeta}{\rho}) + c_2(x)(\frac{x-\zeta}{\rho})^2 + ... \tag{4.5}$$

which can be expressed as the following series

$$C(x,x-\zeta) = \sum_{\xi=0}^{\infty} c_{\xi}(x)(\frac{x-\zeta}{\rho})^{\xi} \tag{4.6}$$

The c_{ξ} are coefficients. These coefficients can be found by enforcing the reproducing conditions or the different moments of the weight function. The Taylor-Young expansion of a C^n continuous function is of the form

$$u(x_i) = u(x) + u'(x)(\frac{x_i-x}{\rho}\rho) + \frac{u''(x)}{2!}(\frac{x_i-x}{\rho}\rho)^2\rho^2 + ... + \frac{u^n(x)}{n!}(\frac{x_i-x}{\rho}\rho)^n\rho^2 \tag{4.7}$$

The discrete form of the corrected kernel approximation of the function at x is written

$$f^h(x) = \sum_{i=1}^{n} \tilde{W}(x-x_i)f_i\Delta x_i \tag{4.8}$$

Substituting the equation 4.7 we obtain

$$\begin{aligned} f^h(x) = &\left(\sum_{i=1}^{n} \tilde{W}(x-x_i)\Delta x_i\right) f(x)h^0 - \left(\sum_{i=1}^{n}(\frac{x-x_i}{h})\tilde{W}(x-x_i)\Delta x_i\right) f'(x)h + ... \\ &+ \left(\sum_{i=1}^{n}(-1)^n(\frac{x-x_i}{h})^n\tilde{W}(x-x_i)\Delta x_i\right) \frac{f(x)}{n!}h^n + O(h^{n+1}) \end{aligned} \tag{4.9}$$

In order to obtain an *n*-th reproducing capability, the moment of the $n-th$ kernel has to be enforced

$$\begin{cases} M_0(x) = \sum_{i=1}^{n} \tilde{W}(x-x_i)\Delta x_i = 1; \\ M_1(x) = \sum_{i=1}^{n} \left(\frac{x-x_i}{h}\right) \tilde{W}(x-x_i)\Delta x_i = 0; \\ \quad \vdots \\ M_n(x) = \sum_{i=1}^{n} \left(\frac{x-x_i}{h}\right)^n \tilde{W}(x-x_i)\Delta x_i = 0; \end{cases} \tag{4.10}$$

By replacing the expression of the corrected kernel 4.4 into equation 4.10, we obtain a system of n+1 equations, which, once resolved, leads to the b_i coefficient of the correction function 4.5.

$$\begin{pmatrix} m_0(x) & m_1(x) & \cdots & m_n(x) \\ m_1(x) & m_2(x) & \cdots & m_{n+1}(x) \\ \vdots & \vdots & \vdots & \vdots \\ m_n(x) & m_{n+1}(x) & \cdots & m_{2n}(x) \end{pmatrix} \begin{pmatrix} b_0(x) \\ b_1(x) \\ \vdots \\ b_n(x) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{4.11}$$

The correction function can also be used to control the influence of the internal nodes for the computation on the boundaries. This technique allows the adherence to the essential boundaries in a more efficient way.

4.2.3 Moving Least Squares (MLS)

The idea of using the moving least squares method as an approximant was first introduced by Lancaster and Salkauskas in 1981 [146] and later by Nayroles [201]. The MLS generalizes the Shepard fitting method [259](see 4.40) to the higher order shape functions. Duarte and Oden first used the interpolant within the framework of the Hp-Cloud approach [75, 76]. Belytschko subsequently applied the similar interpolant within the Galerkin framework giving rise to the element free Galerkin method (EFG) [280]. In the MLS approximation scheme, the approximated function is expressed as follows:

$$u^h(x) = \sum_{i=1}^{n} p_i(x)a_i(x) = P^T(x)a(x) \tag{4.12}$$

Where n is the number of terms in the basis and $p_i(x)$ are the monomial basis functions. $P^T(x) = [p_1(x), ..., p_n(x)]$ is a complete monomial basis of order n. $a_i(x)$ represents the unknown coefficients which are functions of the spatial coordinates. The basis can be either linear or quadratic and singular functions can also be added for problems, which produce singular solutions. The $a_i(x)$ coefficients are obtained by minimising the following L_2 norm at each point of interest x_Q:

$$J(x) = \sum_{i=1}^{n} w_i(x)[u^h(x, x_i) - u(x_i)]^2 \tag{4.13}$$

$$J(x) = \sum_{i=1}^{n} w_i(x)[P^T(x_i)a(x) - u(x_i)]^2 \tag{4.14}$$

Where $w_i(x) = w(x - x_i) > 0$ is the moving least squares weight function with a compact support associated with the node i In a matrix notation

$$J(x) = (Pa - u)^T W(Pa - u) \tag{4.15}$$

then the minimisation problem can be solved by finding extremes of the quadratic form J

$$\frac{\partial J}{\partial a} = 2(P^T WPa - P^T Wu) = 2(A(x)a(x) - B(x)u) = 0 \tag{4.16}$$

with

$$u = [u_1, ..., u_k] \tag{4.17}$$

A(x) is the moment matrix

$$A(x) = \sum_{i=1}^{n} p^T(x_i) w_i(x) p(x_i) \tag{4.18}$$

$$B(x) = w_i(x) p(x_i) \tag{4.19}$$

$$P = \begin{bmatrix} p_1(x_1) & \cdots & p_n(x_1) \\ \vdots & \ddots & \vdots \\ p_1(x_k) & \cdots & p_n(x_k) \end{bmatrix} \tag{4.20}$$

and

$$w_i(x) = \begin{bmatrix} w(x-x_1) & 0 & \cdots & 0 \\ 0 & w(x-x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w(x-x_k) \end{bmatrix} \tag{4.21}$$

Hence it gives for the coefficients a(x)

$$a(x) = A^{-1}(x) B(x) u \tag{4.22}$$

By replacing the expression of the coefficients with the expression of the approximated field function we obtain the following expression

$$u^h(x) = \sum_{i=1}^{n} \phi_i u_i \tag{4.23}$$

$$\Phi(x) = p^T(x) A^{-1}(x) B_i(x) \tag{4.24}$$

Where $\Phi(x) = [\phi_1, ..., \phi_n(x)]$ is the vector containing the MLS shape function for each of the n nodes inside the support domain of the point of interest x. Looking at the above expressions, we can note that, in the case where the weight functions $W_i(x)$ are taken as constants over each local domain, the formulation leads to the FEM. The high order partial derivatives of the MLS shape function can be easily derived from the expression 4.24. For simplicity, we use the subscript i, and the summation over repeated indices to denote the partial spatial derivatives. Using the chain rule, the first order partial spatial derivative of the MLS shape function can be obtained using

$$\Phi_{,i}(x) = p^T_{,i}(x) A^{-1}(x) B(x) + p^T(x) A^{-1}_{,i}(x) B(x) + p^T(x) A^{-1}(x) B_{,i}(x) \tag{4.25}$$

Or rewritten at the node of interest I [280])

$$\phi_{I,i} = \sum_{j}^{N} \left[p_{j,i} (A^{-1} B)_{jI} + p_j \left(A^{-1}_{,i} B + A^{-1} B_{,i} \right)_{jI} \right] \tag{4.26}$$

The problem with the above expression is the presence of the spatial derivative of the inverse of the weight moment matrix $A^{-1}_{,i}$. Considering the matrix A invertible we have

$$A^{-1} A = Id \tag{4.27}$$

Taking the partial derivative of it

$$(A^{-1}A)_{,i} = A^{-1}_{,i}A + A^{-1}A_{,i} = Id_{,i} = 0 \tag{4.28}$$

i.e

$$A^{-1}_{,i}A = -A^{-1}A_{,i} \tag{4.29}$$

Then multiplying by A^{-1}

$$A^{-1}_{,i} = -A^{-1}A_{,i}A^{-1} \tag{4.30}$$

This expression of $A^{-1}_{,i}$ is not suitable as it is computationally expensive which slows down the computation of the spatial derivative of the approximated function. In order to avoid the computation of such expression, Belytschko [29] proposed the following method to derive and speed up the computation of spatial derivatives for MLS shape without having to calculate the inverse of the A matrix. Taking

$$\gamma^T = p^T A^{-1} \tag{4.31}$$

Which allow to write

$$\Phi^T(x) = \gamma^T(x)B(x) \tag{4.32}$$

$\gamma(x)$ being symmetric, it can be obtained with a linear solver from

$$A\gamma = p \tag{4.33}$$

Then the derivative of $\Phi^T(x)$ can be obtained using

$$\Phi^T_{,i} = \gamma^T_{,i}B + \gamma^T B_{,i} \tag{4.34}$$

$$\Phi^T_{,ij} = \gamma^T_{,ij}B + \gamma^T_{,i}B_{,j} + \gamma^T_{,j}B_{,i} + \gamma^T B_{,ij} \tag{4.35}$$

$$\Phi^T_{,ij} = \gamma^T_{,ijk}B + \gamma^T_{,ij}B_{,k} + \gamma^T_{,ik}B_{,j} + \gamma^T_{jk}B_{,i} + \gamma^T_{,i}B_{,jk} + \gamma^T_{,j}B_{,ik} + \gamma^T_{,k}B_{,ij} + \gamma^T B_{,ijk} \tag{4.36}$$

Where partial derivatives of γ can be obtained from the following linear algebraic equations

$$A\gamma_{,i} = p_{,i} - A_{,i}\gamma \tag{4.37}$$

$$A\gamma_{,ij} = p_{,ij} - (A_{,i}\gamma_{,j} + A_{,j}\gamma_{,i} + A_{,ij}\gamma) \tag{4.38}$$

$$A\gamma_{,ijk} = p_{,ijk} - (A_{,i}\gamma_{,jk} + A_{,j}\gamma_{,ik} + A_{,k}\gamma_{ij} + A_{,ij}\gamma_k + A_{,ik}\gamma_j + A_{,jk}\gamma_i + A_{,ijk}\gamma) \tag{4.39}$$

The Shepard interpolant [259] developed at the origin for surface representation can be considered as a particular case of the MLS. In fact if we take the lowest order polynomial basis $p^t(x) = \{1\}$, the MLS shape function for the point of interest x_q is

$$\Phi^T_J(x_q) = \frac{W(x_q - x_J)}{\sum_I W(x_q - x_I)} \tag{4.40}$$

Another particular form is the one used in the diffuse element method. It is very much similar to the EFG as it uses the MLS approximant. The DEM was introduced by Nayroles [201], The DEM

method uses a truncated form for the computation of the derivatives of the approximated function

$$u^h(x) = \sum_{i=1}^{n} p_i(x)a_i(x) = p^T(x)a(x) \tag{4.41}$$

Using the chain rules, the first order derivative is given by

$$\frac{\partial u^h(x)}{\partial x_j} = \frac{\partial P^T(x)}{\partial x_j}a(x) + \underbrace{p^T(x)\frac{\partial a(x)}{\partial x_j}}_{Truncated\ by\ the\ DEM\ method} \tag{4.42}$$

From equation 4.24 we can rewrite the derivative of the MLS SF as

$$\frac{\partial \Phi_i(x)}{\partial x_j} = p^T A^{-1}(x)B(x) \tag{4.43}$$

4.2.3.1 Weight functions

The MLS method requires the use of weight function in its formulation. The role of the weight function is very important as it controls the "smoothness" of how nodes enter and leave the support domain Ω_s as the point of interest is moving. The continuity of the MLS Shape function is controlled by the continuity of the basis function $p^T(x)$ and the smoothness of the weight function. In practice, a low order-basis is usually preferred due to constraints imposed on the number of nodes to include in the support domain. The rank of the Moment matrix A should be n to guarantee its invertibility. Under such conditions, it is even more obvious that the smoothness of the MLS shape function is dependent on the weight function's higher order of continuity.
The weight function $W(x - x_I) = W_i$ defined over the support domain Ω_s should fulfil certain properties [187] to guarantee the compatibility similar to the ones of the kernels for the SPH. It should vanish outside Ω_s and be strictly positive over the support domain Ω_s. It has to also be smooth and monotonically decreasing from the center of the domain. The high degree of continuity provided by the weight function over its support domain Ω_s, is a criterion of the compatibility of the MLS shape function over the entire domain Ω. Here is a list of some of the more commonly used weight functions.

Exponential

$$W_I(x) = \begin{cases} e^{-(\frac{d_I}{c})^2} & d_I \leq 1 \\ 0 & d_I > 1 \end{cases} \tag{4.44}$$

or has suggested by be Belytschko [280]

$$W_i(x) = \begin{cases} \frac{e^{-(\frac{d_I}{c})^{2k}} - e^{-(\frac{d_W}{c})^{2k}}}{1 - e^{(\frac{d_I}{d_W})^{2k}}} & d_I \leq 1 \\ 0 & d_I > 1 \end{cases} \tag{4.45}$$

The parameter c controls the shape of the exponential weight function. When it decreases, the weight increases on the point located close to x_I. Belytschko defines $c = \alpha c_I$ with $1 \leq \alpha \leq 2$ and

$$c_I = \overset{max}{j \in S_J} \|x_j - X_I\| \tag{4.46}$$

S_J is described as the minimum set of neighbouring points of x_I which constructs a polygon

around the point of interest (quadrature point) x_I. In the case of a regular distribution of nodes, C_I implies the maximum distance between two nodes of the support domain. In the case of an irregular distribution, c_I corresponds to the characteristic length of the integration zone for the point x_I.

Conical

$$W(x) = \begin{cases} 1 - (d_I)^{2K} \leq 1 \\ 0 \qquad\qquad d_I > 1 \end{cases} \tag{4.47}$$

Cubic spline

$$W(x) = \begin{cases} \frac{2}{3} - 4d_I^2 + 4d_I^3 & d_I \leq 0.5 \\ \frac{4}{3} - 4d_I + 4d_I^2 - \frac{4}{3}d_I^3 & 0.5 < d_I \leq 1 \\ 0 & d_I > 1 \end{cases} \tag{4.48}$$

Quartic spline

$$W(x) = \begin{cases} 1 - 6d_I^2 + 8d_I^3 - 3d_I^4 & d_I \leq 1 \\ 0 & d_I > 1 \end{cases} \tag{4.49}$$

Where d_I is the relative distance between the point of interest x and the node x_i.

$$d_I = \frac{|x - x_i|}{d_W} \tag{4.50}$$

Here d_W refers to the size of the weight function domain. d_W works as a scaling parameter for the size of the subdomain Ω_I around the point of interest x_I. For instance the weight function domain is usually the same as the support domain but can be different as in the case of the local weak formulation (see section 5.2). It does not need to be constant for all the nodes of domain Ω. r_i is the normalized radius of influence which defines the domain of influence of each integration point. This domain of influence corresponds to the support of the weight function for the reason given previously. A natural choice is to choose the weight function providing the higher order of continuity such as the quartic spline.

As shown in figure 4.4, all the weight functions are smooth and provide first order continuity. One can observe that both the quartic spline and the cubic spline are very close in shape. The new quartic weight function possesses second order reproducing capacity.

The first derivatives of all the weight function are smooth and the cubic spline and quartic spline are very close in shape. The second derivative of the cubic spline is not smooth anymore.

The derivatives at all orders of the exponential weight function 4.44 are continuous in the support domain. However the exponential weight function and all its derivatives remain non-zero on the boundaries of their support domain. Hence it should prevent the exponential weight function to provide any order of compatibility. In practice it is not the case and a high order of compatibility is obtained as long as the support domain defined is sufficiently large. The second derivative of the new quartic weight function shows the same non-zero characteristic on its boundaries.

To illustrate this we consider a square domain $\Omega(x, y) \in [-1, 1] \times [-1, 1]$ discretised by 11×11 fields nodes uniformly distributed. The shape functions are evaluated using a 51×51 uniformly distributed sampling points. The MLS shape functions and its derivatives evaluated at the location $x = (0, 0)$. are shown on figure 4.5 and 4.6.

As long as the weight function and its derivatives are continuous and positive up to the desired degree, the choice of the weight function remains quite arbitrary. For practical reasons, one-piece weight functions like the quartic spline are usually preferred. Because the spline function is of a

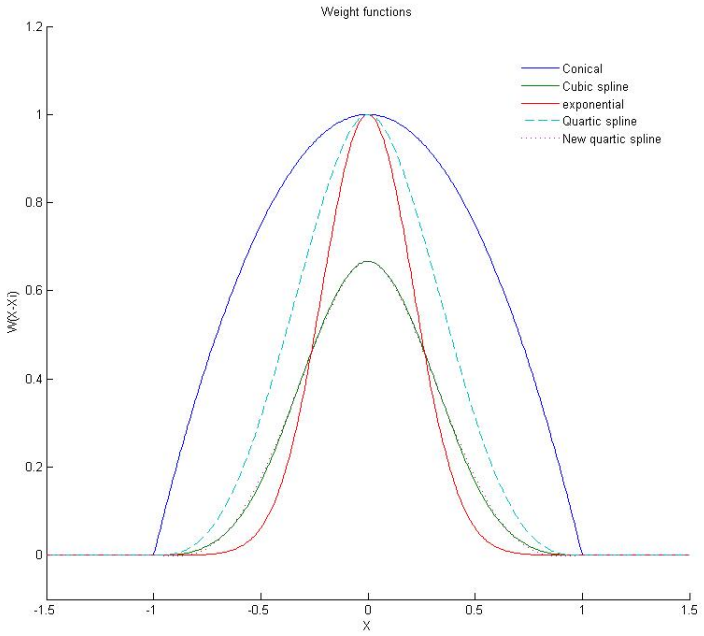


Figure 4.4: 1-dimensional Weight functions

polynomial form, the shape function built with the spline function as a weight function possesses the characteristic of the rational function. By carefully choosing the order of continuity of the spline function, one can build C^k continuous shape functions. There is an easy way of building such a type of weight function with the desired order of continuity [7, 161]. Hence by using this technique, it is fairly easy to build a shape function $\Phi(x) \in C^k$ with k as the order desired. The quartic spline in equation 4.49 is an example of a spline built with the technique. The following examples of weight function are built similarly.

- Quartic spine: 2^{rd}order continuity

$$W_I(x) = \begin{cases} 1 - 6d_I^2 + 8d_I^3 - 3d_I^4 & d_I \leq 1 \\ 0 & d_I > 1 \end{cases} \tag{4.51}$$

- 7^{th} order spline function with 3^{th} order continuity

$$W_I(x) = \begin{cases} 1 - 35d_I^4 + 84d_I^5 - 70d_I^6 + 20d_I^7 & d_I \leq 1 \\ 0 & d_I > 1 \end{cases} \tag{4.52}$$

- 9^{th} order spline function with 4^{th} order continuity

$$W_I(x) = \begin{cases} 1 - 126d_I^5 + 420d_I^6 + -540d_I^7 + 315d_I^8 - 70d_I^9 & 0 \leq d_I \leq 1 \\ 0 & d_I > 1 \end{cases} \tag{4.53}$$

In practice, the choice of the weight function is determined by the order of continuity desired. The MLS shape function is complete up to the order of the basis . The smoothness of the nodal shape function $\Phi(x)$ is determined by the choice of the weight function. The choice of the weight function is arbitrary as long as the weight function is positive and continuous. However, assuring

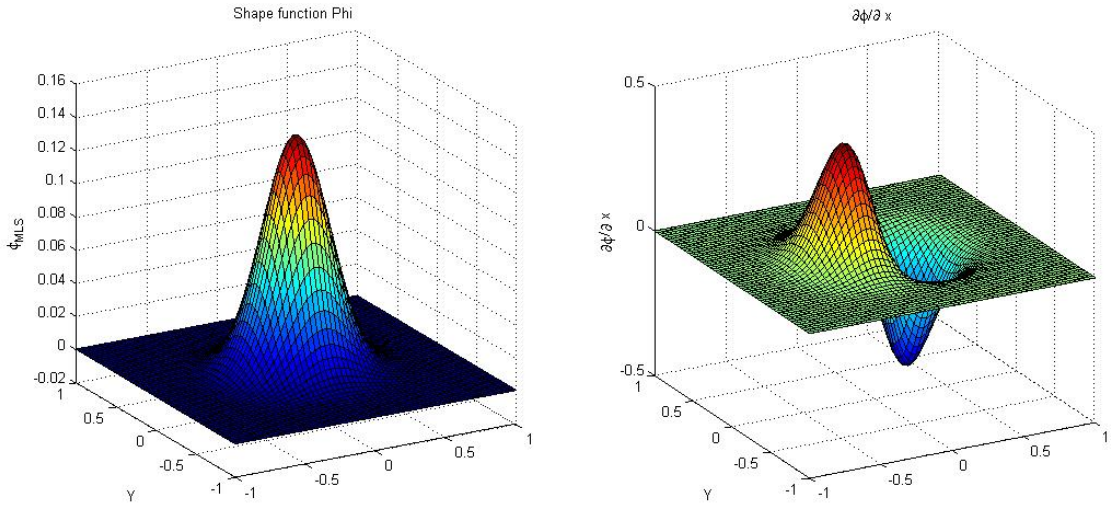


Figure 4.5: MLS shape function and first derivative at sampling point x = (0,0) using 11×11 field nodes

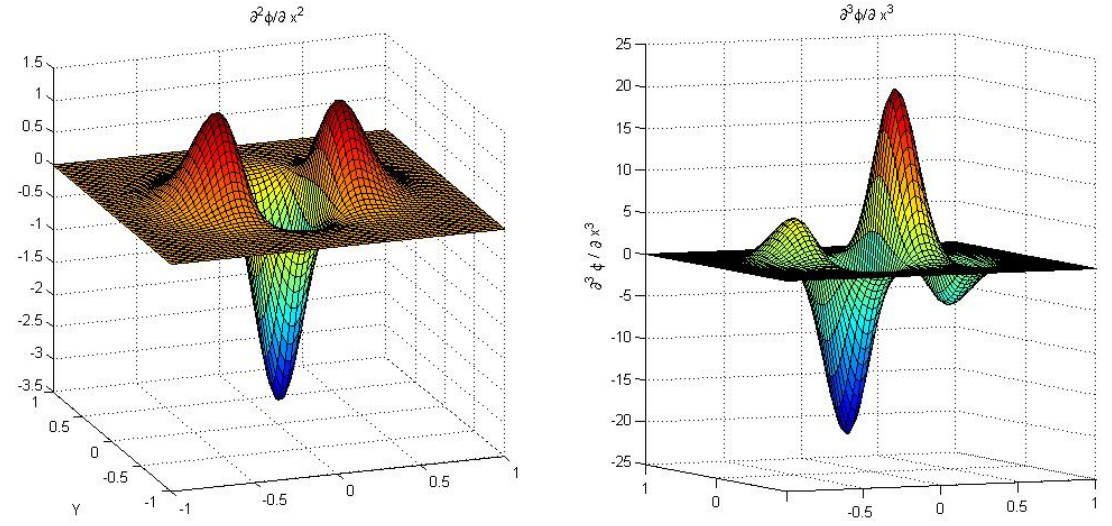


Figure 4.6: MLS shape function second and third derivatives at sampling point x = (0,0) using 11×11 field nodes

a certain smoothness in the shape function derivatives is important because discontinuities may result in difficulties in the numerical integration of the shape functions.

4.2.4 Radial basis functions based methods

Multiquadrics also known as radial basis functions (RBF) for the numerical solution of PDEs have first been used in the method of fundamental solution dual reciprocity method (MFS-DRM). This method was derived from the dual reciprocity boundary element method (DRBEM) but used the method of fundamental solution (MFS) [81,143] instead of the BEM for the discretisation process. MFS is a meshless technique for the numerical solution of certain elliptic boundary value problems, which fall in the class of the methods generally called boundary methods. Like the boundary element method (BEM), it is applicable when a fundamental solution of the differential equation in question is known. It shares the same advantages as the BEM over domain discretisation methods, but also has certain advantages over the BEM. The use of fictitious boundaries outside of domain of interest allows us to get rid of the singularity present in the BEM formulation.

MFS is a truly meshless method, which does not require a domain or boundary discretisation. As a collocation based method, integration over element domain is not needed. Thus the method does not suffer from any increase in the dimension of the problem and can benefit when used in high dimensional problems. Being used only on homogeneous problems at first, the MFS method regained more attention since its successful application for non-homogenous problems and time dependent problems [214] using the dual reciprocity method (DRM) coupling [200]. In the BEM, the DRM allows us to transfer the domain integration on to the boundary domain. DRM is a class of boundary element techniques; the domain integral resulting from non-homogeneous terms in Poisson type equations is transferred to equivalent boundary integrals by using suitable approximation functions. The use of RBF as approximating functions for this purpose has several advantages over conventional interpolation techniques. The MFS is proven to converge exponentially for smooth shape and boundaries.

Another direction in the use of RBF in the meshfree method is the one developed by Kansa [132]. Since its introduction in early 90's, RBFs have attracted increasing attention for their use as interpolants in meshfree methods as well as radial basis networks for machine learning [248]. Kansa was the first to use the RBFs as an interpolant within a collocation type formulation [132, 133] followed by Schaback [246] and Wendland [295] who introduced the RBF in the same formulation as the EFG.

Radial functions are a special class of function with the characteristic feature that their response decreases (or increases) monotonically with distance from a central point. The center, the distance scale, and the precise shape of the radial function are parameters of the model, which are all fixed if it is linear. The definition of a RBF is a univariate function $g : \mathbb{R}^d \to \mathbb{R}$ with $d \geq 1$. It can be written $g(r)$ with $r = |x|$ being the Euclidean norm.

Consider $x_1, ..., x_N \in \Omega \subset \mathbb{R}^n$ be a given set of nodes. Let

$$g_j(x) \equiv g(\|x - x_j\|) \in \mathbb{R} \quad j = 1, ..., N \tag{4.54}$$

be a set of any radial basis functions. The norm $\|x - x_j\|$ considered here is the Euclidean distance. Given the interpolation data values $u_1, ..., u_N \in \mathbb{R}$ at the locations $x_1, ..., x_N \in \Omega \subset \mathbb{R}^n$, the RBF interpolant is

$$F(x) = \sum_{j=1}^{N} \alpha_j g_j(x) + \alpha_N + 1 \tag{4.55}$$

It is obtained by solving the system of N+1 linear equations

$$\sum_{j=1}^{N} \alpha_j g_j(x_i) + \alpha_N + 1 = u_i \quad \forall i \in [1; N] \tag{4.56}$$

$$\sum_{j=1}^{N} \alpha_j = 0 \tag{4.57}$$

The formulation was first introduced by Hardy [107, 108]

$$p = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^N, \tag{4.58}$$

And

$$G = \begin{bmatrix} g_1(x_1) & \dots & g_N(x_1) \\ \vdots & \ddots & \vdots \\ g_1(x_N) & \vdots & g_N(x_N) \end{bmatrix} \in \mathbb{R}^{N\times N}, H = \begin{bmatrix} G & p \\ p^T & 0 \end{bmatrix} \in \mathbb{R}^{N+1\times N+1}, \tag{4.59}$$

The equation 4.56 can be expressed in matrix form

$$H\alpha = y \tag{4.60}$$

With $\alpha = (\alpha_1, \dots, \alpha_{N+1})^T, y = (u_1, \dots, u_N, 0)^T \in \mathbb{R}^{N+1}$ From the previous equation the set of coefficients can be obtained by

$$\alpha = H^{-1}y \tag{4.61}$$

The derivatives of the interpolant at the nodes x_i can be obtained by

$$F'(x_i) = \sum_{j=1}^{N} \alpha_j g_j'(x_i) \quad \forall i \in [1, N] \tag{4.62}$$

And for the n-derivatives we have

$$F^n(x_i) = \sum_{j=1}^{N} \alpha_j g_j^n(x_i) \quad \forall i \in [1, N] \tag{4.63}$$

The formulation is the following for the approximated function at the point of interest.

$$u(x_Q) = \sum_{i=1}^{n} R_i(x)a_i(x_Q) = R^T(x)a(x_Q) \tag{4.64}$$

R_i is the radial basis function. The only variable is the distance $r = \|x_k - x_i\|$ between the point of interest x and a node at x_i. The vector R(x) contains the radial basis functions at each node contained in the support domain. Enforcing u(x) to pass through each scattered node included in the support domain of the point of interest x_Q gives the set of coefficients a_i. The a_i coefficients are given by

$$a = R_Q^{-1}U_S \tag{4.65}$$

U_S is the vector containing all the field nodal variables at the n nodes included in the support domain. Substituting the expression of the a_i coefficients into the expression of the approximated field function u gives

$$u = \sum_{i=1}^{n} R_i(x)a_i(x_Q) = R^T(x)a(x_Q) = R^T(x)R_Q^{-1}U_S \tag{4.66}$$

Here are some comon RBFs [30, 108]:
Multi-quadrics (MQ) RBF

$$R(x, y) = (r^2 + (\alpha_c d_c)^2)^Q \tag{4.67}$$

Gaussian(EXP)

$$R(x, y) = e^{\left[-\alpha_c\left(\frac{r}{d_c}\right)\right]} \tag{4.68}$$

Linear splines

$$R(x,y) = r \tag{4.69}$$

Thin Plate Spline(TPS)

$$R(x,y) = r^n \quad \text{or} \quad R(x,y) = r \times ln(r) \tag{4.70}$$

Logarithmic

$$R(x,y) = r^n log(r) \tag{4.71}$$

α_c and Q, correspond to the dimensionless shape parameters of the RBF. d_c is the characteristic length which can be defined from the average nodal spacing inside the support domain of a point x_Q. Radial basis function methods are originally non-local, which do not lead to sparse stiffness matrices. To solve those problems, compactly supported RBF(CSRBF) have been developed [49, 50, 294, 298].

CSRBF Wendland-C2 [118]

$$R(x,y) = (1 - \frac{r}{\delta})^4(1 + 4\frac{r}{\delta}) \tag{4.72}$$

CSRBF Wendland-C4 (Wen-C4)

$$R(x,y) = (1 - \frac{r}{\delta})^6(3 + 18\frac{r}{\delta} + 35(\frac{r}{\delta})^2) \tag{4.73}$$

CSRBF Wendland-C6 (Wen-C6)

$$R(x,y) = (1 - \frac{r}{\delta})^8(1 + 8\frac{r}{\delta} + 25(\frac{r}{\delta})^2 + 32(\frac{r}{\delta})^3) \tag{4.74}$$

CSRBF Wu-C2 [298]

$$R(x,y) = (1 - \frac{r}{\delta})^5(8 + 40\frac{r}{\delta} + 48\frac{r^2}{\delta^2} + 25\frac{r^3}{\delta^3} + 5\frac{r^4}{\delta^4}), \tag{4.75}$$

CSRBF Wu-C4

$$R(x,y) = (1 - \frac{r}{\delta})^6(6 + 36\frac{r}{\delta} + 82\frac{r^2}{\delta^2} + 72\frac{r^3}{\delta^3} + 30\frac{r^4}{\delta^4} + 5\frac{r^5}{\delta^5}), \tag{4.76}$$

CSRBF Buhmann C2 (Buh-C2) [51]

$$R(x,y) = 1 - 12\frac{r^2}{\delta^2} + 32\frac{r^3}{\delta^3} - 21\frac{r^4}{\delta^4} + 12r^4 logr \tag{4.77}$$

RBFs are defined over the whole domain ω, which is a subset of $\mathbb{R}^n$ whereas the CSRBF are only defined over a local compact support with a parameter δ which controls the size of the compact domain.

Figure 4.7 and 4.8 illustrates the RPIM-MQ shape function and its derivatives. 51×51 sampling points are used to plot the shape functions with a regular 5×5 distribution of field nodes to build the SF. Q=0.5 and $\alpha_c = 2.0$ For comparison the Wen-C4 and its derivatives are shown on on figure 4.9.

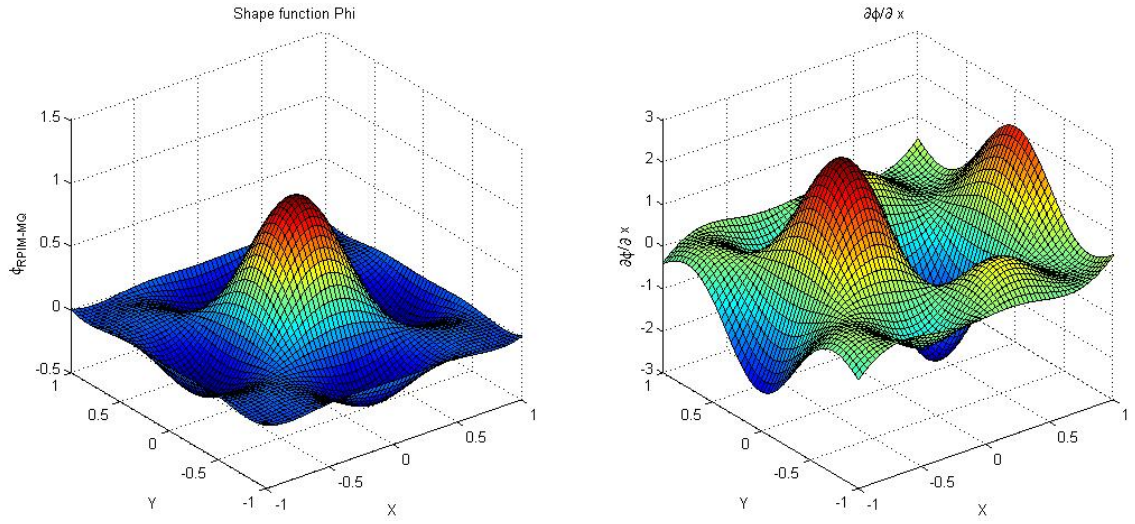


Figure 4.7: RPIM-MQ shape function and first derivative at x = (0,0) for a 5×5 regular distribution of nodes

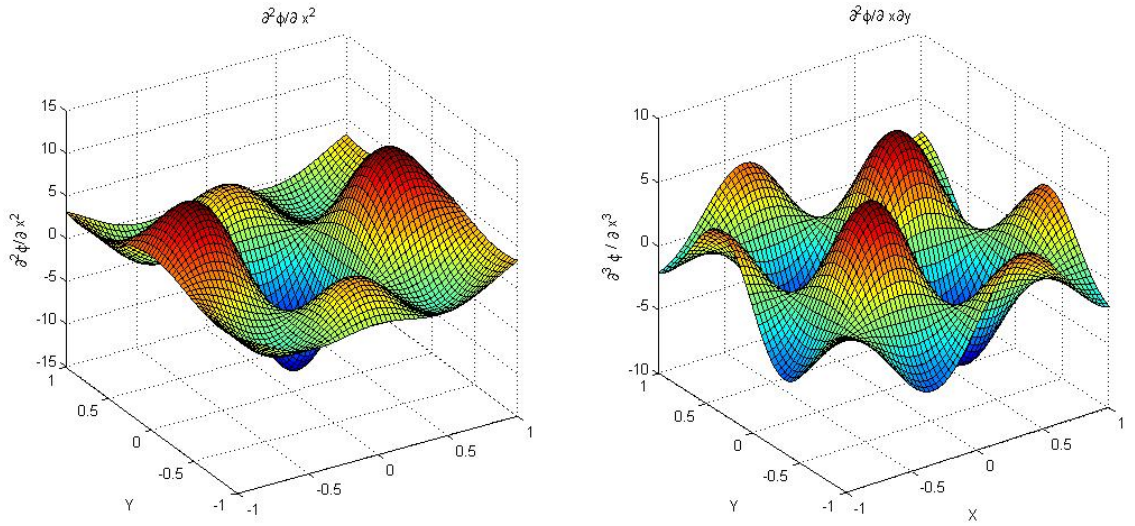


Figure 4.8: Second Derivative and composed derivative of the RPIM-MQ shape function at x=(0,0) for a 5×5 regular distribution of nodes

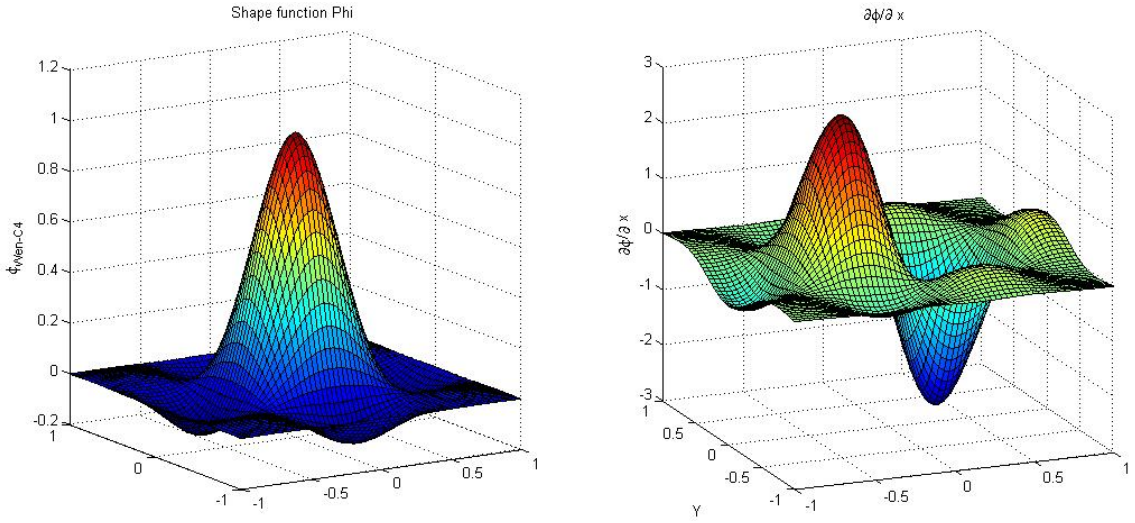


Figure 4.9: RPIM-Wendland C4 shape function and first derivative at x = (0,0) for a 5×5 regular distribution of nodes. $\theta = d_c$.

Error estimates for RBF approximation on scattered data can be found in the Schaback and Wu's publication [247, 299].

4.2.5 Point Interpolation Method (PIM)

The main idea behind PIM [159]is to make the interpolation function pass through the function values at each node within the support domain (cf MLS). The PIM belongs to the finite series representation method. It interpolates the function u(x) using values of each node included in the support domain4.3 of the point of interest x_Q. The finite series formulation gives the interpolated function

$$u^h(x) = \sum_{i=1}^{m} B_i(x) a_i(x_Q) \tag{4.78}$$

Where B_i(x) are the basis functions defined in the Cartesian coordinates space x= (x, y, z). Here m is the number of basis functions and $a_i(x_Q)$ is the set of unknown coefficients associated with the point x_Q . n corresponds to the number of nodes in the local support domain of the point of interest x_Q. This support domain can be either circular or rectangular (Figure4.3).In the case of PIM we have n=m to make SFs passing though the function values at each node in the support domain. Here the basis functions are polynomial kind. The formulation for the approximated function at a point of interest x_Q is the following

$$u^h(x) = \sum_{i=1}^{n} p_i(x) a_i(x_Q) = [p_1(x)...p_m] \left\{ \begin{array}{c} a_1 \\ \vdots \\ a_m \end{array} \right\} = p^T(x) a(x_Q) \tag{4.79}$$

p_i(x) is the basis of monomials. For 2D problem the basis is built over Pascal triangles (pascal pyramid for 3D).

$$p^T(x, y) = \{1, x, y, xy, x^2, y^2, x^2y, xy^2, ..., x^n, y^n\} \tag{4.80}$$

The a_i(x) coefficients can be determined by enforcing u(x) to pass through the nodal values at the n nodes included in the support domain of the point of interest x_Q.

$$\begin{array}{c} u_1 = \sum_{i=1}^{m} a_i p(x_1) = a_1 + a_2 x_1 + a_3 y_1 + \cdots + a_m p_m(X_1) \\ \vdots \\ u_n = \sum_{i=1}^{m} a_i p(x_n) = a_1 + a_2 x_n + a_3 y_n + \cdots + a_m p_m(X_n) \end{array} \tag{4.81}$$

In the matrix form, we obtain the following system

$$U = P_m a \tag{4.82}$$

P_m is called the moment matrix and is given by

$$P_m = \left[\begin{array}{cccccc} 1 & x_1 & y_1 & x_1 y_1 & \cdots & p_m(x_1) \\ 1 & x_2 & y_2 & x_2 y_2 & \cdots & p_m(x_2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & y_n & x_n y_n & \cdots & p_m(x_n) \end{array} \right] \tag{4.83}$$

By substituting the expression of the unknown coefficients into the approximated function formulation given below, we obtain

$$u^h(x) = \sum_{i=1}^{n} \phi(x) u_i = \Phi(x) U_s \tag{4.84}$$

Where $\Phi(x)$ is the matrix containing the PIM shape functions

$$\Phi(x) = p^T(x) P_Q^{-1} = [\phi_1(x), \phi_2(x), ..., \phi_n(x)] \tag{4.85}$$

Because all the functions involved in the calculation of the shape function are polynomials, the k^{th} derivatives of PIM shape functions can be obtained by

$$\Phi^{(k)}(x) = \left[p^T(x)\right] P_Q^{-1} \tag{4.86}$$

A major potential problem is the singularity of the moment matrix 4.83. This problem appears in the PIM for certain configuration of nodes. It can be solved by using radial basis functions as in the RPIM. Liu [159] also introduced the MTA as well as nodal perturbation as a way to avoid this problem. However it leads to more effort for the implementation of the method as well as an increase in the computational cost.

4.3 Interpolations accuracy of meshless shape functions

In the following sections, we study the comportment of MLS and RPIM approximants for surface data fitting to optimise their use in future meshfree applications. The test are implemented in Matlab code.

4.3.1 RBF parameter and interpolation performance of the RPIM

We study the effect of the different RBF and CSRBF parameters on the RPIM interpolation accuracy for the fitting of 2D surfaces. We consider the domain $\Omega : \forall(x, y) \in [0, 10] \times [0, 10]$. We use 100 interpolation points x_q distributed evenly such as 10×10. An example is shown on figure 4.10, where the circle represents field nodes and + represents x_q.

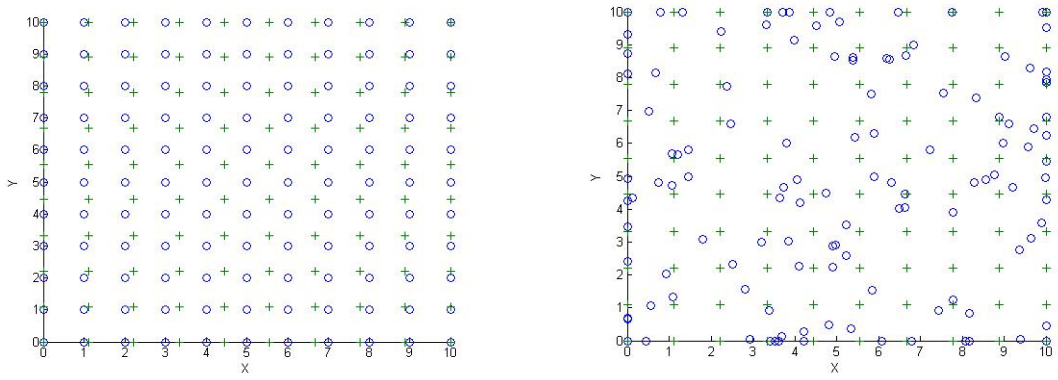


Figure 4.10: Regular and irregular 121 field nodes (o) and regular distribution 100 sampling points x_q(+)

The choice of the topology of the distribution of the sampling points x_q does not really matter

as long it does not coincide with the distribution of the field nodes. For the reproducibility of the results, we use regular distribution of x_q which does not match the position of the field nodes. Both regular and irregular pattern of field nodes are used for the tests. As an indicator of interpolation performance, we use the average fitting error norm indicator given for the whole domain by

$$e^l = \frac{1}{N} \sum \frac{|(u_i^h)^l - u_i^l|}{|u_i^l|} \tag{4.87}$$

With N being the number of sampling points. u_i is the exact function value at the point i and u_i^h corresponds to the approximated function. *l* refers to the derivative order of the function. The first order derivative of the fitted function is given for *l*=1 and so on for the higher order derivatives. The following 2D non polynomial and non rational function is considered as fitted function

$$u(x, y) = a + \sin(kx)\cos(ky) \tag{4.88}$$

With a, k $\in \mathbb{R}$. We set $k = \frac{\pi}{6}$ and $a = 3.5$ for our test. A comparison between the analytical and the interpolated solution for the function and its first derivative is showsn respectively on figures 4.11 and 4.12.

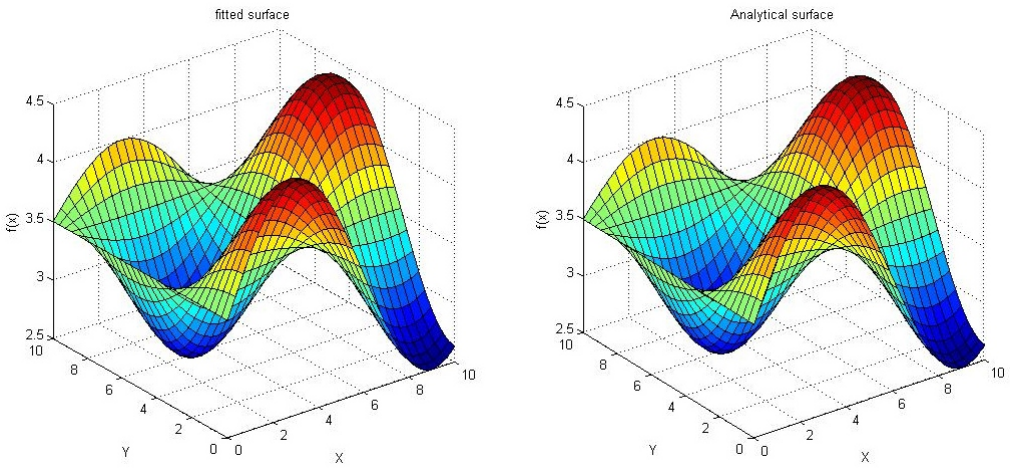


Figure 4.11: Analytical surface vs interpolated surface using 400 irregular nodes and 31×31 sampling points. Error = 1.5×10^{-5}

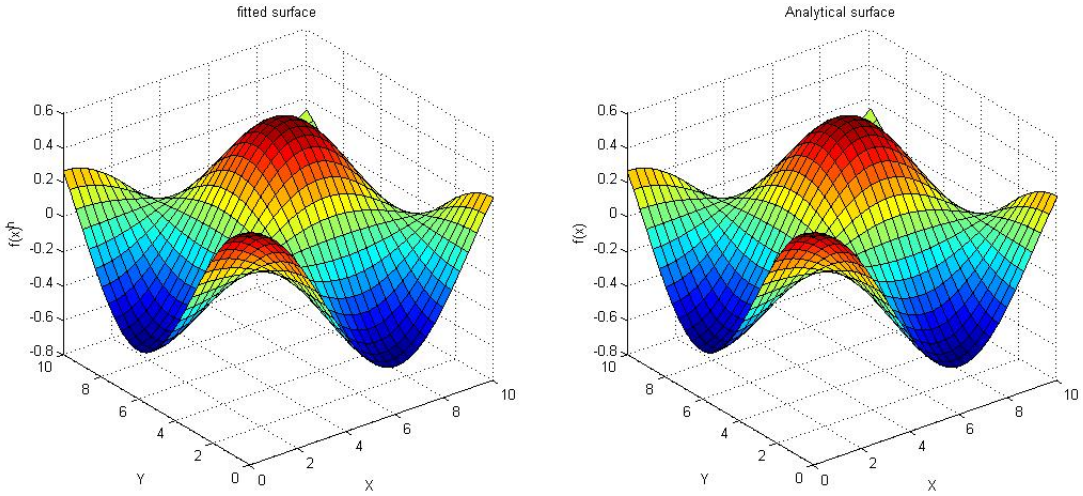


Figure 4.12: Interpolated surface derivative using 400 irregular nodes. Error = 7.1×10^{-6}

The fitted function is smooth and presents enough variation for the proper assessment of the fitting capabilities of the meshfree interpolant. Using the meshfree interpolant, the approximated function can be expressed at the interpolation point x_q by

$$U^h(x_q) = \Phi(x_q)u_i = \sum_{i=1}^{n} \phi_i u_i \tag{4.89}$$

Where ϕ_i and u_i are respectively, the SFs and values of the function associated at each node in the support domain of the point of interest x_q. n refers to the numbers of nodes inside the support domain Ω_q of the point x_q. The derivatives of the approximated function can be approximated by

$$\frac{\partial^l u^h(x_q)}{\partial x^l} = \frac{\partial^l \Phi(x_q)}{\partial x^l} U_i = \sum_{i=1}^{n} \frac{\partial^l \phi_i}{\partial x^l} u_i \tag{4.90}$$

In building RPIM SFs, the computation of the RBFs requires values for the set of parameters. These parameters have an impact on the accuracy of the interpolation and on the conditioning of the matrix R, thus on the computational effort because SFs are computed as many times as there is quadrature points in MFM. The parameter α_S which controls the size of the support domain is set to 3. The size of the support domain d_s for an interpolation point x_q is given by $d_s = \alpha_s d_c$. d_c refers to the average nodal spacing. In the case of the RPIM MQ, α_c and Q are the two parameters which control the performance of the approximation.

We study the influence of the parameter Q on the interpolation accuracy of the RPIM. Set the parameters α_c= 3.0 and $k = 5$. The error for different Q values is plotted on figure 4.13. The blue and green lines are the fitting errors respectively for the function and the first partial derivative. Figure 4.13 on the left shows the impact of Q on the condition number of the moment matrix.

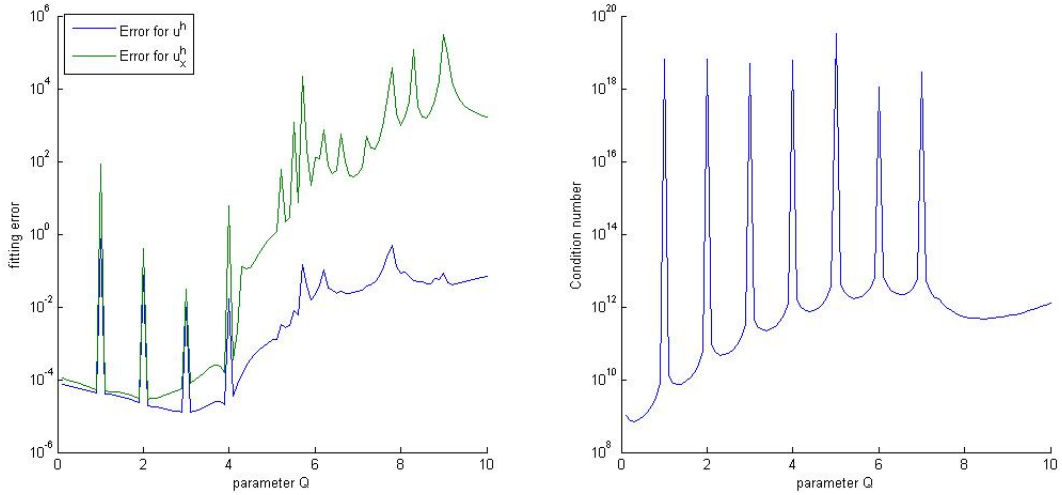


Figure 4.13: Influence of parameter Q on the error on the surface fitting using RPIM-MQ shape function and condition number of the R matrix. 10×10 regular nodes distribution. α_s=3.0, $\alpha_c = 3.0$.

The accuracy is the highest when the parameter $Q \in$ to the interval [0;4]. A closer view on figure 4.14 shows that the RPIM-MQ fitting presents the particularity to be more accurate when the moment matrix is close to be singular. It is clear that the parameter Q influences the accuracy of the RPIM interpolation. Best values for Q to be considered are the ones very close to the values of Q for which the moment matrix is ill conditioned, such as 1.0, 2.0, 3.0. For this reason, closed

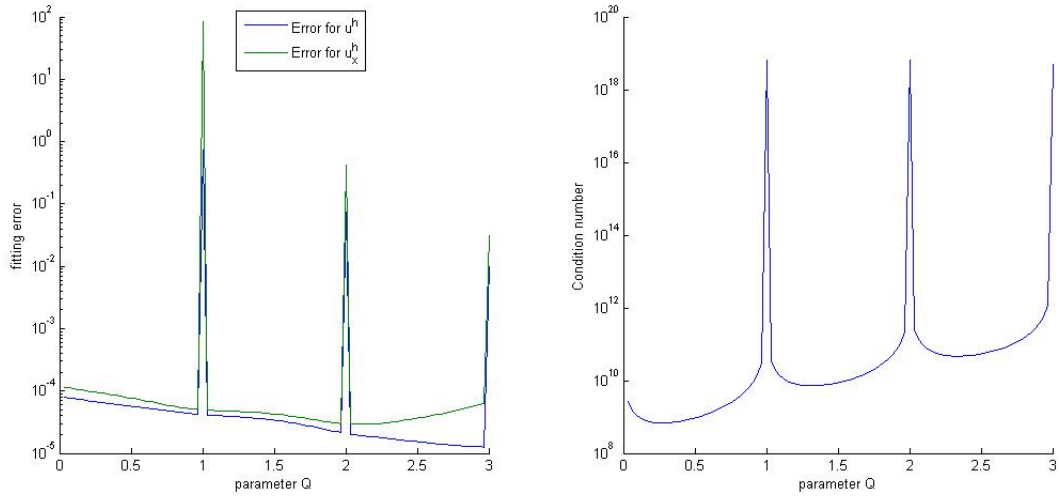


Figure 4.14: Close up view on the influence of parameter Q on the error on the surface fitting using RPIM-MQ shape function and condition number of the R matrix. 10×10 regular nodes distribution. $\alpha_s = 3.0$, $\alpha_c = 3.0$.

values such as 0.95, 1.02 or 1.94 can provide optimised accuracy while maintaining acceptable conditioning for single or double floating point precision.

The influence of the parameter α_c on the RPIM-MQ fitting accuracy and the condition number of the moment matrix is shown in figure 4.15 for a regular 21×21 distribution of nodes. Blue, green and red lines show the error respectively for the function, the 1^{st} and 2^{nd} partial derivative along x coordinate.

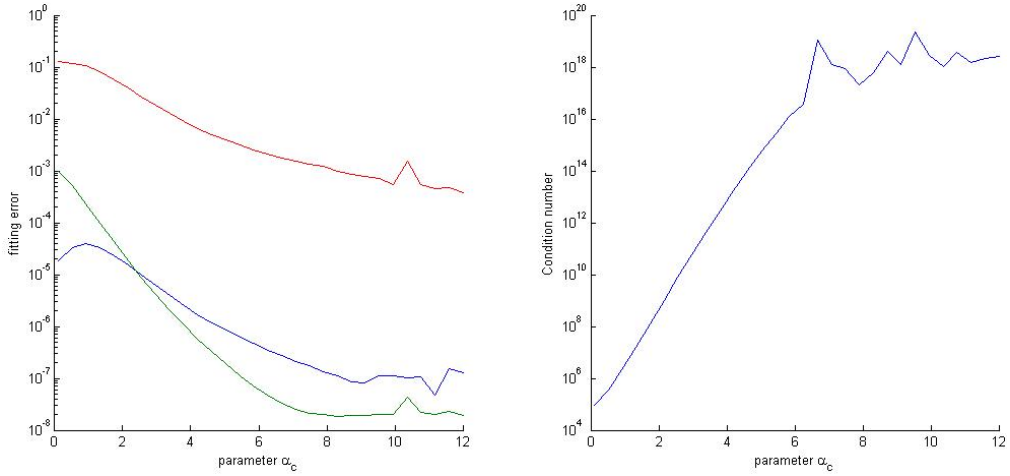


Figure 4.15: Influence of parameter α_c on the fitting error(left) and the condition number(right) of the moment matrix, using RPIM-MQ shape function. $\alpha_s = 4.0$, $q = 0.95$

The parameter α_c influences the fitting accuracy of the RPIM as shown in figure 4.15. The conditioning of the moment matrix degrades as α_c increases. For RPIM-MQ, we will choose values of $\alpha_c \in$ [3;6] for the best compromise between conditioning and performance. A similar result is obtain for 441 irregular distributed nodes as shown on figure 4.16.

For the RPIM with Gaussian RBF the values of parameter α_c are quite different. As shown on the figure 4.17 the smaller α_c, the better the accuracy and higher the condition number . In order to control the computational effort, values of α_c must be chosen properly to control the conditioning number of the moment matrix. Using a refined range of values (figure 4.18) for α_c shows that

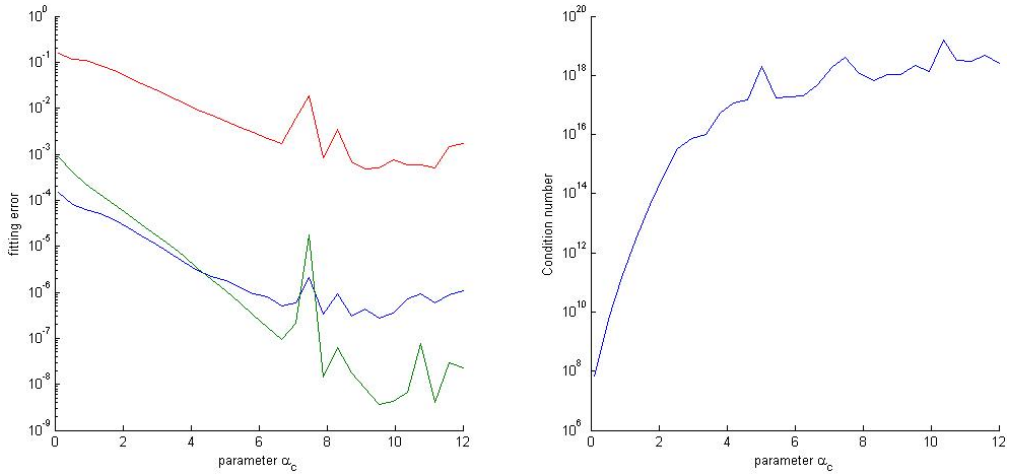


Figure 4.16: Influence of parameter α_c on the error in surface fitting using RPIM-MQ shape function. $\alpha_s = 4.0$, $q = 0.95$. 441 irregular filed nodes

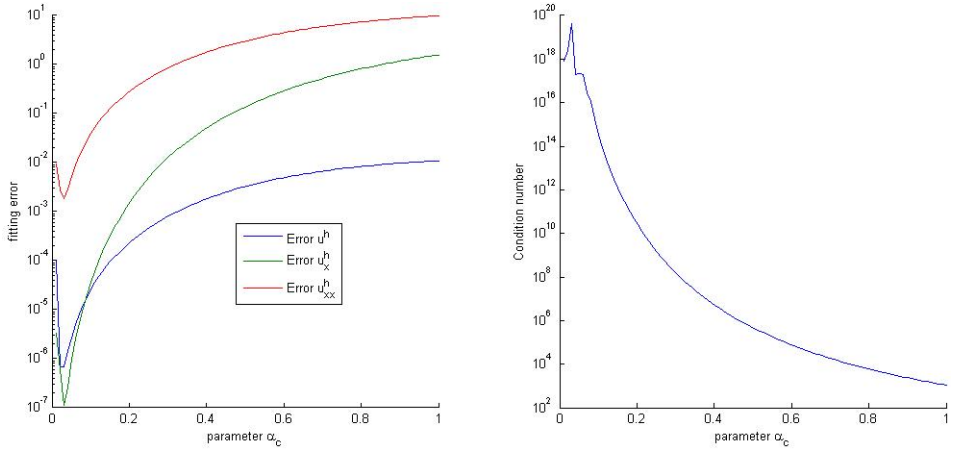


Figure 4.17: Influence of parameter α_c on the error in surface fitting using RPIM-EXP shape function. $\alpha_s = 4.0$

values $\in [0.001\ 0.05]$ offer great trade-off between accuracy and computational cost. For practical problems we will use values $\alpha_c = 0.01, 0.02, 0.06$.

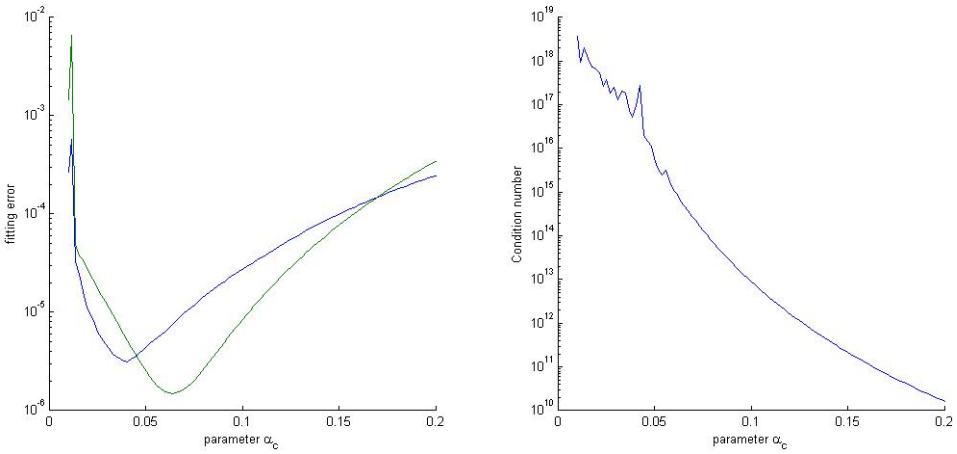


Figure 4.18: Influence of parameter α_c on the fitting error and condition number using RPIM-EXP shape function. α_s = 4.0. 121 irregular nodes distribution

The results of our test agree with existing literature for practical engineering problems [236, 293].

It is always better to fine-tune the parameters based on a pre-study of the benchmark problem close to the practical one.
In the case of CSRBF, the choice of the parameter $\delta = \alpha_c \times d_c$ control the local support. Figure 4.19 shows the influence of the parameter on few Wendland and Wu's CSRBF on the fitting error for the function and the first partial derivative along the x coordinate.

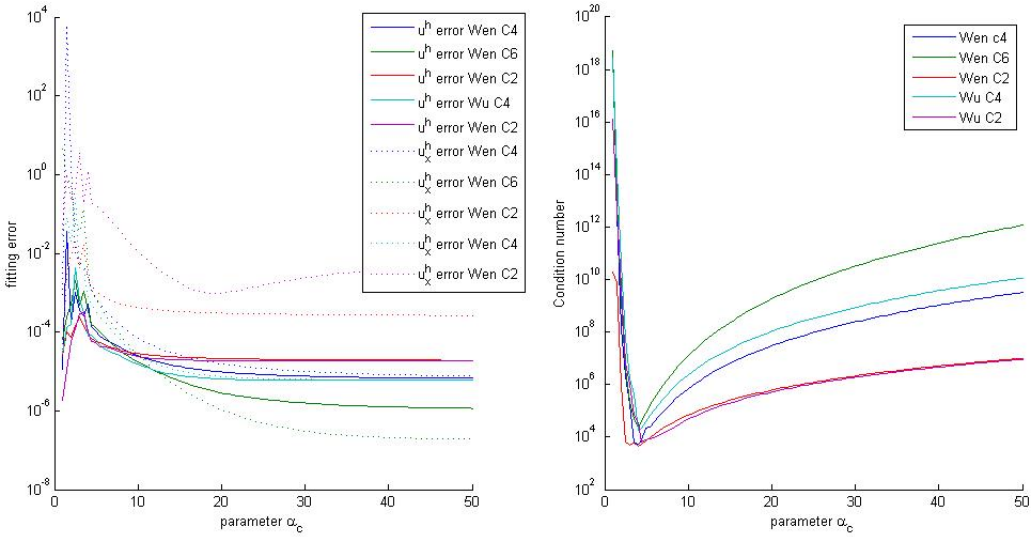


Figure 4.19: Influence of parameter α_c on the error in surface fitting and condition number of the RPIM matrix using RPIM Wendland CSRBF shape function. α_s = 4.0

For CSRBF, the wider the local support size the more accurate the approximation. Increasing the size of the local support domain increases the condition number of the RPIM moment matrix but not in a drastic manner. Apart from smoothness, the degree of the CSRBF does not impact the accuracy much.

4.3.1.1 Influence of support domain

We study the influence of the size of the support domain on the fitting accuracy of the RPIM SF as shown on figure 4.20. As the size of the support domain increase the condition number of the moment matrix increase also. Figure 4.21 shows how varying the support domain size impacts the average number numbers of nodes included in the support domain and the computational time for the whole approximation procedure.
A similar result obtained for CSRBF time was shown on figure 4.22 and 4.23 for the Wendland WEN-C6.

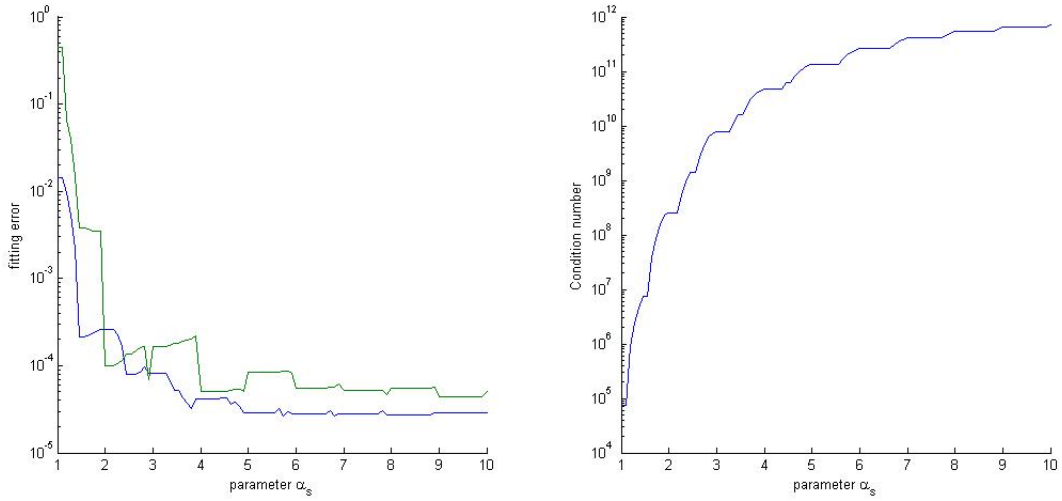


Figure 4.20: Influence of parameter α_s on the error in surface fitting(left) and the condition number of the R matrix(right) using RPIM-MQ shape function. α_c=3, $q = 0.95$.

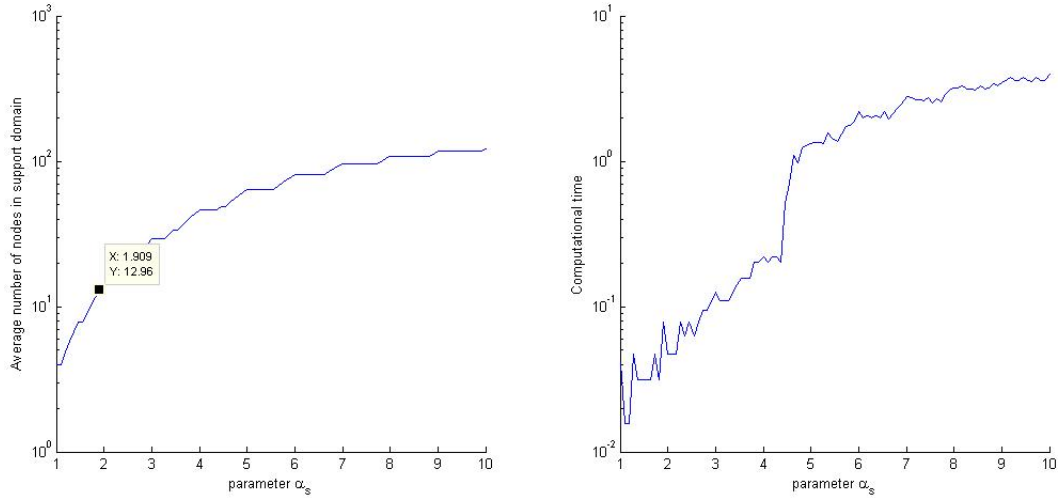


Figure 4.21: Influence of α_s on the computational time and the average number of nodes included in the support domain.

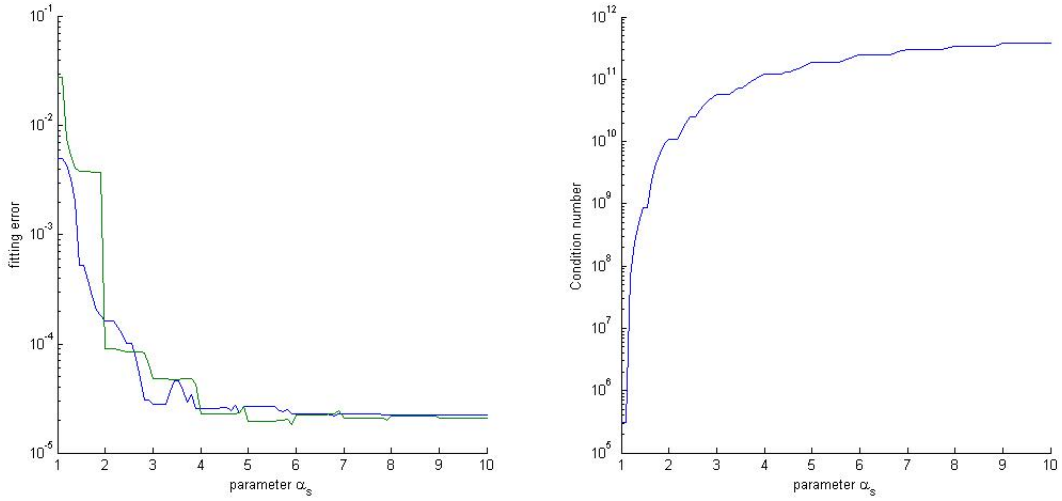


Figure 4.22: Influence of parameter α_s on the error in surface fitting using RPIM-Wen C6 shape function. α_c=40

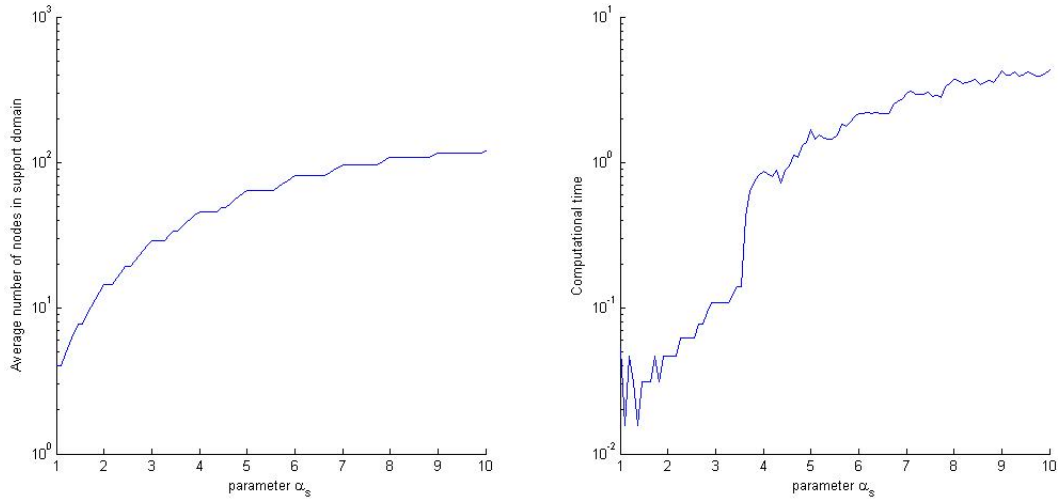


Figure 4.23: Influence of α_s on the computational time and the average number of nodes included in the support domain.

It appears that the RPIM works well for a support domain with $\alpha_s \in [2, 4]$ which corresponds to a number of nodes of $\approx [10, 45]$ included in the support domain. It provides a good trade off between accuracy, reasonable conditioning of the RPIM moment matrix and computational cost. We will usually carry our test with $\alpha_c = 4$.

4.3.1.2 Convergence study

To study the influence of h refinement on the fitting accuracy of RPIM shape functions, we consider the domain Ω of study such as $\forall(x, y) \in [0; 10] \times [0; 10]$ and the 2D function given by equation 4.89 as the function to interpolate, defined over Ω. Regularly distributed points are used as sampling points. The refinement h refers to the average nodal distance d_c between the nodes within a support domain. We take the following refinement of the constellation of nodes

$$h = \frac{L}{N_s} \tag{4.91}$$

L corresponds to the characteristic length of the domain Ω and N_d is the average density of nodes. For a regular distribution of nodes, it is the number of nodes along one side. This definition is similar to the h-convergence one uses for mesh techniques such as FEM. We use a 12×12 regular distribution of x_q, q = 0.95 and $\alpha_s = \alpha_c = 3.0$. The error for the RPIM-MQ for the fitting of u(x) and its first derivative for both regular(left) and irregular distribution(right) of nodes are shown on figure 4.24. Figure 4.25 shows the impact of the refinement of h on the condition number of the moment matrix for both regular (left) and irregular (right) nodal distribution. Increasing the density of nodes through the paramter h have a similar effect as what is otbtained by increasing α_s. However the accuracy provided by the h refinement seems to be better than the one by enlarging support domain.

The choice of α_c is affected by the fill distance. For instance, for the RPIM-EXP, as the density is refined, the optimal value of α_c decreases. This is highlighted in figure 4.26 and figure 4.27.

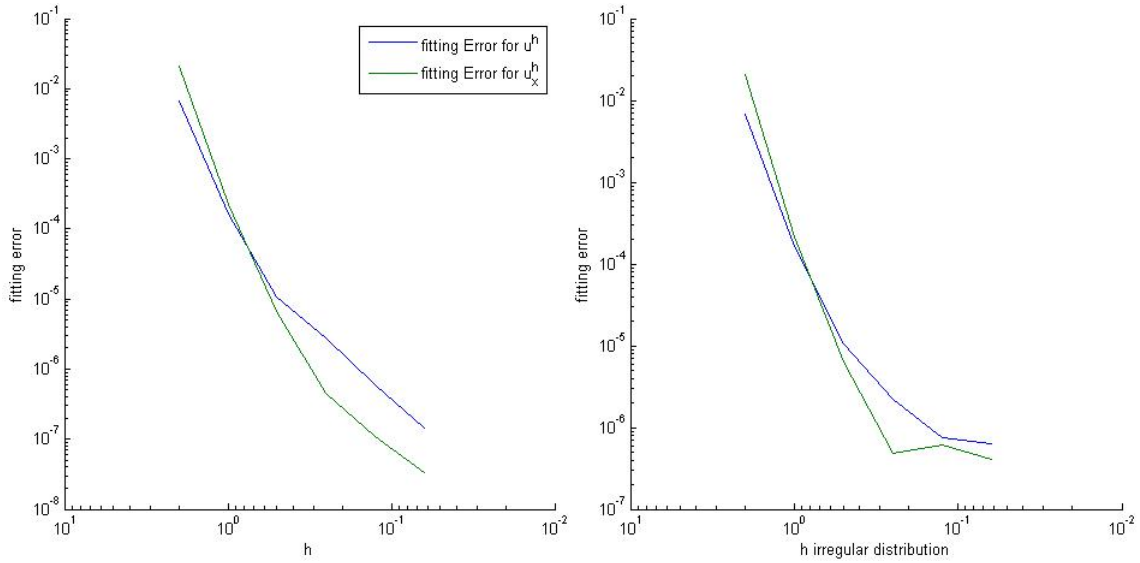


Figure 4.24: Convergence of the error in surface fitting using RPIM-MQ SF. $\alpha_s = 3.0$, $\alpha_c = 3.0$, $q = 0.95$

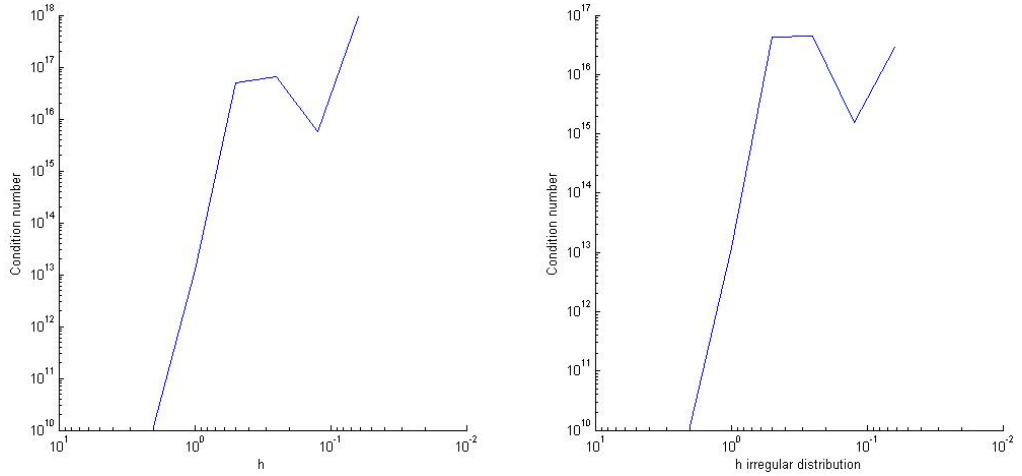


Figure 4.25: Influence of h on the conditioning of the moment matrix of the RPIM-MQ SF for regular and irregular distribution of nodes $\alpha_s = 3.0$,$\alpha_c = 3.0$,$q = 0.95$

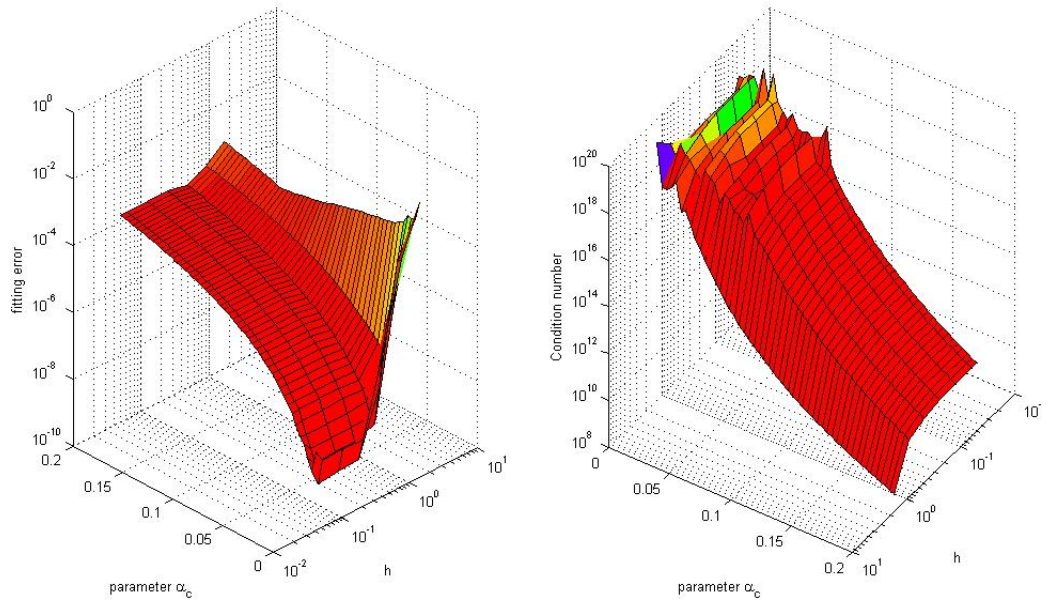


Figure 4.26: Influence of h and α_c on the fitting error and the conditioning of the moment matrix of the RPIM-EXP shape function. $\alpha_c = 4.0$

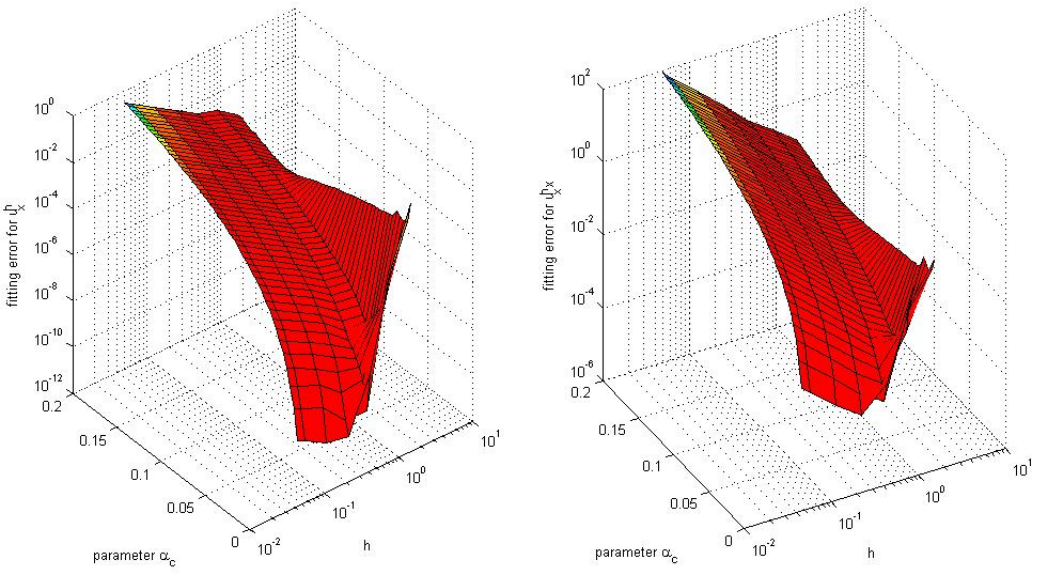


Figure 4.27: Influence of h and α_c on the fitting error for the RPIM-EXP shape function first (left) and second derivative(right). $\alpha_c = 4.0$

For sufficiently smooth functions, the MQ RBF approximation is exponentially or spectrally accurate with an error decaying at the rate $\mathcal{O}(\eta^N)$ where $0 < \eta < 1$. Unlike the traditional mesh based approximation method which exhibits algebraic convergence rates where the decay is $\mathcal{O}(N^{-m})$ for $m \in \mathbb{R}$. For meshfree RBF type of approximation, we distinguish

- Stationary approximation: the number of centers N is fixed and the shape parameters ϵ is refined towards 0 (figure 4.28). Such a convergence is unique to RBF and there are no similar representations in polynomial based methods.
- Non-stationary approximation: The shape parameters are fixed and N is increased in a way which mimics the process of convergence of polynomial based methods (figure 4.29).

Asymptotic approximation properties are quantified by the notion of approximation order. In scattered data RBF, the fill distance, also referred as density, indicates how well the set of centers fills out the domain Ω. From a geometrical standpoint, the fill distance is the radius of largest possible empty ball that can be placed among the centers in the domain. Looking at the MQ, for a fixed shape parameter ϵ, converges at a spectral rate when h decreases ($\equiv N$ increase). We have the following error [48]

$$|f(x) - f^h(x)| \leq e^{\frac{-K(\epsilon}{h}} \tag{4.92}$$

$K(\epsilon)$ is a constant that depends on the value of the shape parameter of the RBF. A problem is quantifying how K varies along with ϵ which makes it inappropriate for stationary convergence assessment [177, 178].

$$|f(x) - f^h(x)| \leq K\eta^{(\frac{1}{\epsilon h})},\ 0 < \eta < 1 \tag{4.93}$$

Spectral convergence is achieved by either h or ϵ converges toward 0. A problem with RBF interpolant is to achieve the numerical theoretical convergence rates as the condition number of the system matrix increases when the shape parameter or h decreases as shown in figure 4.28. We interpolate the simple function $f(x) = e^{(sin(\pi x))}$ over the domain $\Omega = [-1, 1]$. We first study the convergence of the MQ RBF approximation for various shape parameters ϵ. We use 100 nodes evenly spread through the domain Ω.

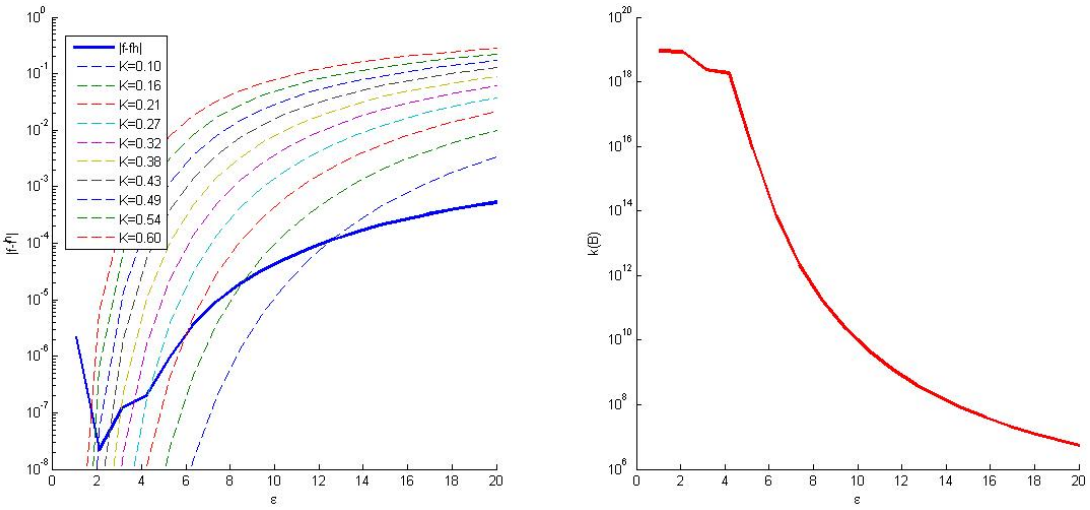


Figure 4.28: Error norm and error bound vs shape parameter (left) . Condition number of the interpolation matrix vs shape parameter (right).

The error estimate in equation 4.93 is plotted in a dash line on figure 4.28. The convergence is achieved and the bound for $K \approx 0.25 - 0.27$ holds, for $\epsilon > 5$. For $\epsilon < 5$ the condition number of the interpolation matrix $k(B) > \mathcal{O}(10^{18})$ which is too big for the 32-bit double float precision to allow accurate results. The desired convergence cannot be obtained then. The optimal value for the shape parameter seems so be somewhat around 5 considering the ill conditioned state of the B matrix for lower ϵ values.
In the non-stationary case, similar results can be observed as the number of nodes N to describe the domain is increased (or the fill distance h decreases).

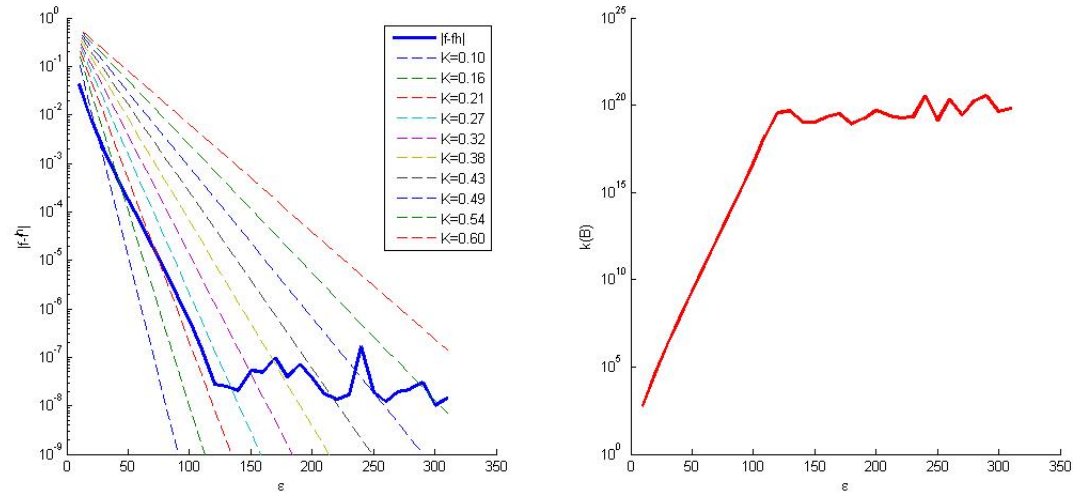


Figure 4.29: Error norm and error bound vs number of nodes (left). Condition number of the interpolation matrix vs number of nodes (right).

The accuracy increases as the total number of nodes used to build the approximation matrix increases. For a number of nodes N>100, the condition number of the interpolation matrix degrades as well. This is an illustration of the classical issue that arises though the use of RBF for building approximant. One problem is finding optimal shape parameters for which we can obtain the most accurate results when solving the system in equation 4.60. There are different possibilities which have been used:

- Using non-standard algorithm to solve the problem (pseudo inverse, pre-conditioner solver etc...). This does not always solve the problem.
- Use augmented precision computer arithmetic.
- Adjusting both N and the shape parameter to take advantage of each impact on the accuracy and condition number of the system matrix to obtain an optimal set of shape parameters.

In previous tests, we chose to keep the condition number of the system matrix in the range $[10^{14}, 10^{17}]$. By playing on both, we have to keep the condition number in the range to obtain the best accuracy. This is referred to as the MQ critically conditioned in literature when the method produces an accurate approximation but not if the conditioned number was increased. Sarra covered this subject with great depth in [245]. However in practice, in meshfree methods it is not a requirement to obtain an optimal convergence rate.

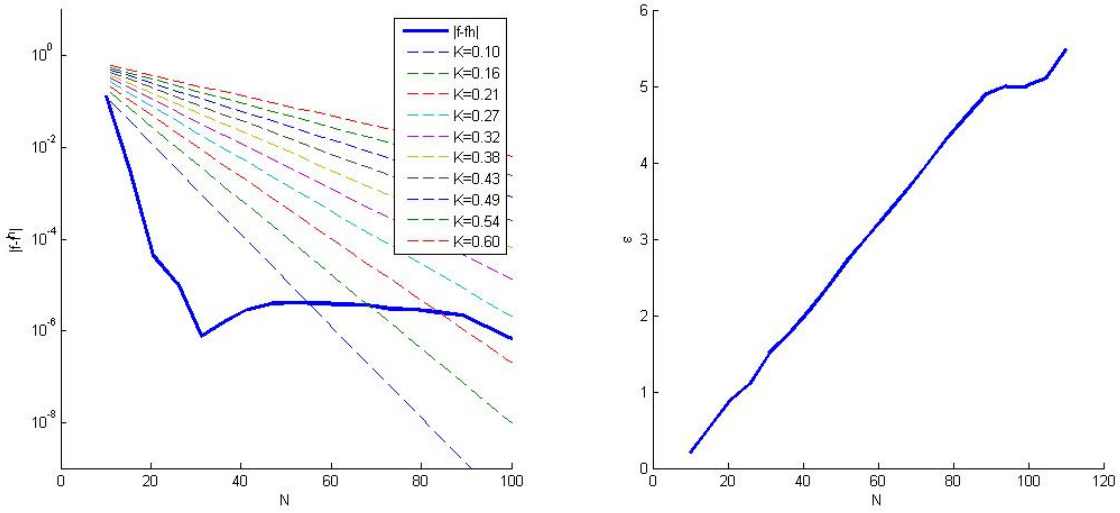


Figure 4.30: Error norm and error bound vs number of nodes (left). Condition number of the interpolation matrix vs number of nodes (right).

4.3.1.3 Polynomial enrichment

Due to its mathematical form RPIM fails to represent the linear field exactly. The standard RPIM without polynomial term fails to pass the standard patch test. This is a major drawback in mechanical problems where the reproduction of linear fields is a condition for the standard patch test. Adding polynomial terms in the construction of the RPIM SF helps to solve this problem. Our test shows that the addition of the linear basis helps to achieve linear reproduction to the machine accuracy as shown in figure 4.31.
The addition of polynomial terms in the basis improves the overall accuracy slightly as shown on figure 4.32 for the RPIM-MQ and figure 4.33 for RPIM-EXP.

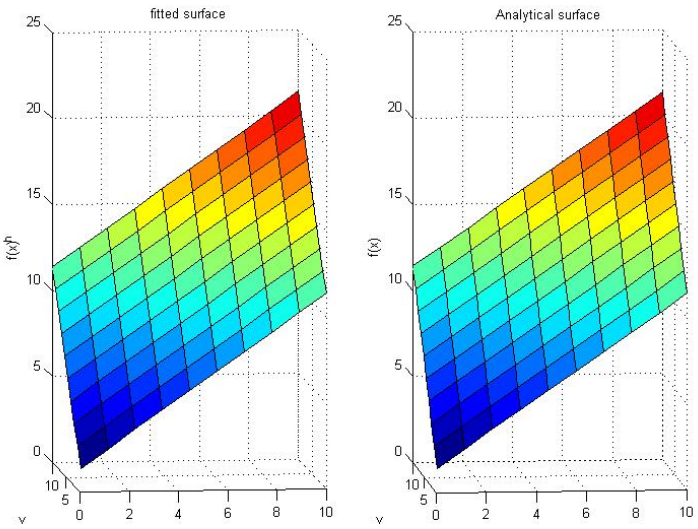


Figure 4.31: Exact linear field reproduction with RPIM-MQ over a 121 irregular distribution of nodes. $\alpha_c = 3.0$, Q = 0.95. Error 4.4×10^{-15}

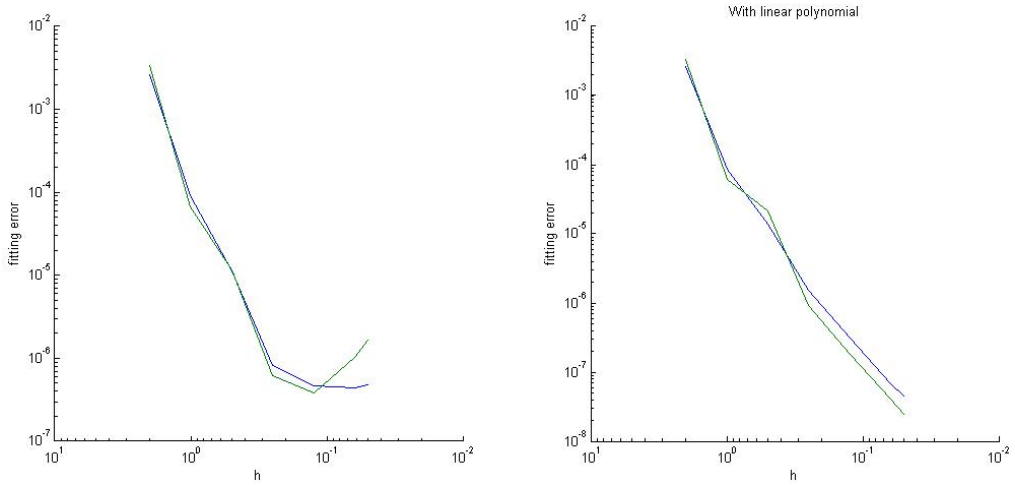


Figure 4.32: Error for different h for RPIM-MQ without polynomial term (left) and with polynomial term (right).$\alpha_c = 3.0$, Q = 0.95.

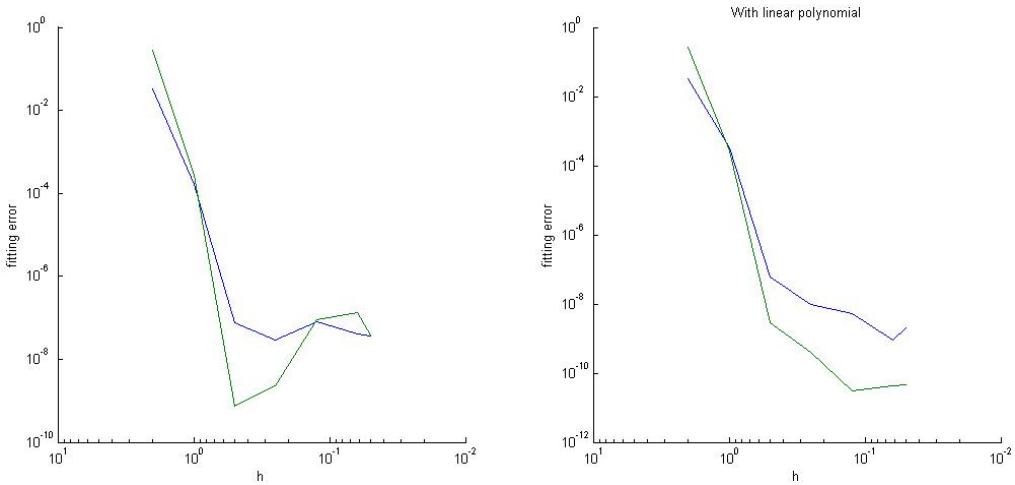


Figure 4.33: Error for different h for RPIM-EXP without polynomial term (left) and with polynomial term (right).Irregular field nodes distribution. $\alpha_c = 0.03$.

Figure 4.34 and 4.35 illustrate that augmenting the basis with linear polynomial does not improve

the conditioning of the moment matrix.

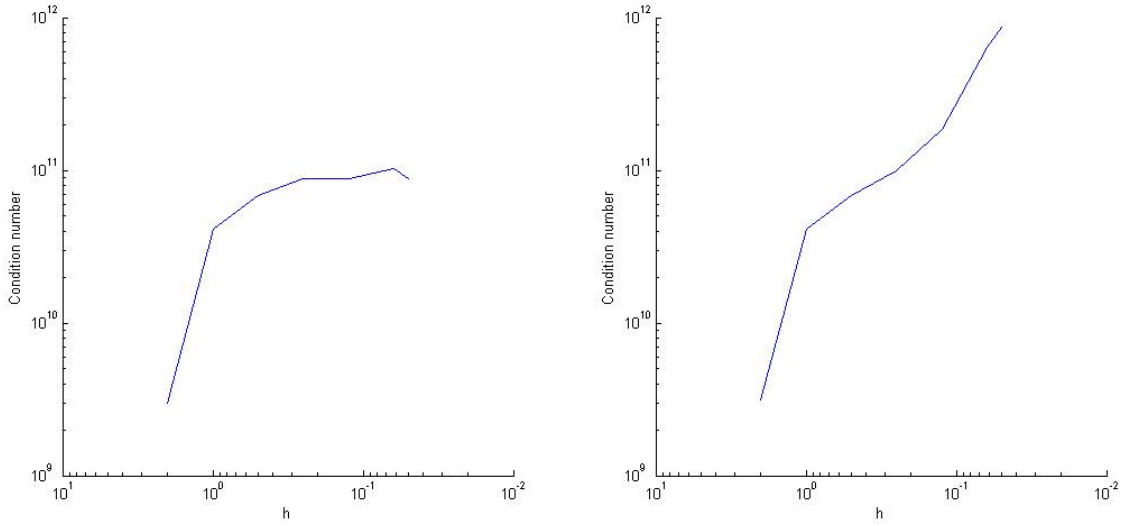


Figure 4.34: Influence of refinement h on the condition number of the moment matrix for RPIM-MQ without polynomial term (left) and with polynomial term (right). $\alpha_c = 3.0$, q = 0.95.

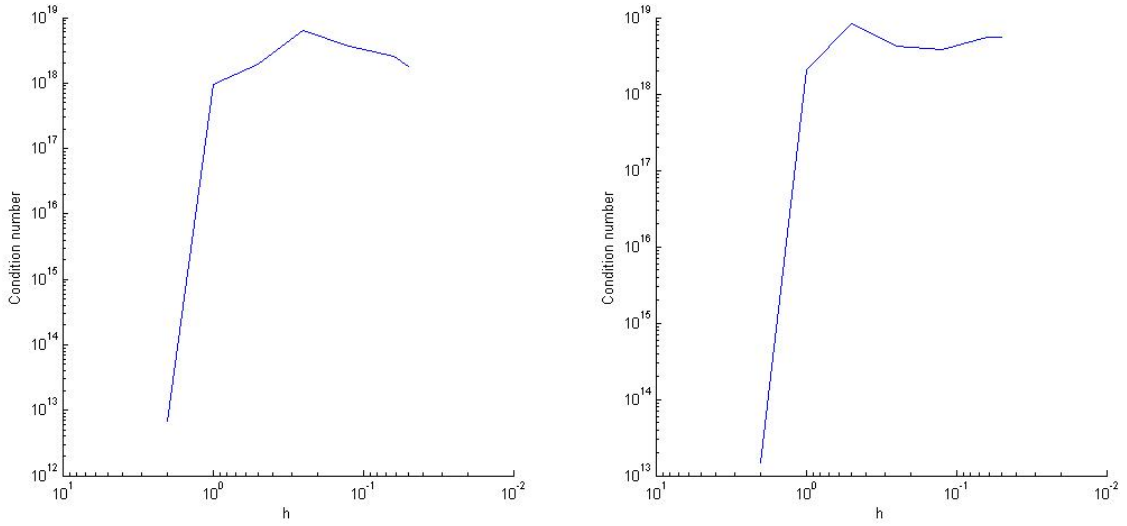


Figure 4.35: Influence of refinement h on the condition number of the RPIM matrix for RPIM-EXP without polynomial term (left) and with polynomial term (right).Irregular field nodes distribution. $\alpha_c = 0.03$.

Figure 4.36 shows the effect of nodal distribution on the convergence and condition number of the moment matrix for RPIM-MQ. Fitting errors for the function, its first and second partial derivative are respectively plotted in blue green and red. Figure 4.37 shows similar results for the RPIM-EXP.

In the case of CSRBF, a similar benefit from adding polynomial terms has also been found. For example, Figure 4.38 shows a comparison of the convergence with and without the use of polynomial terms for the WEN-C6 CSRBF.

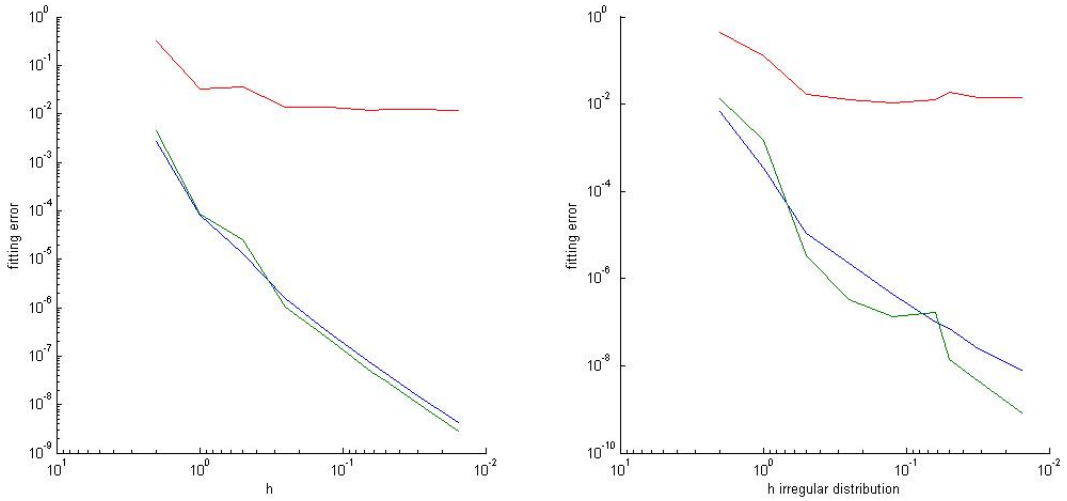


Figure 4.36: Comparison of the fitting error's convergence of RPIM-MQ for regular (left) and irregular (right) nodal distribution. $\alpha_c = 3.0$.

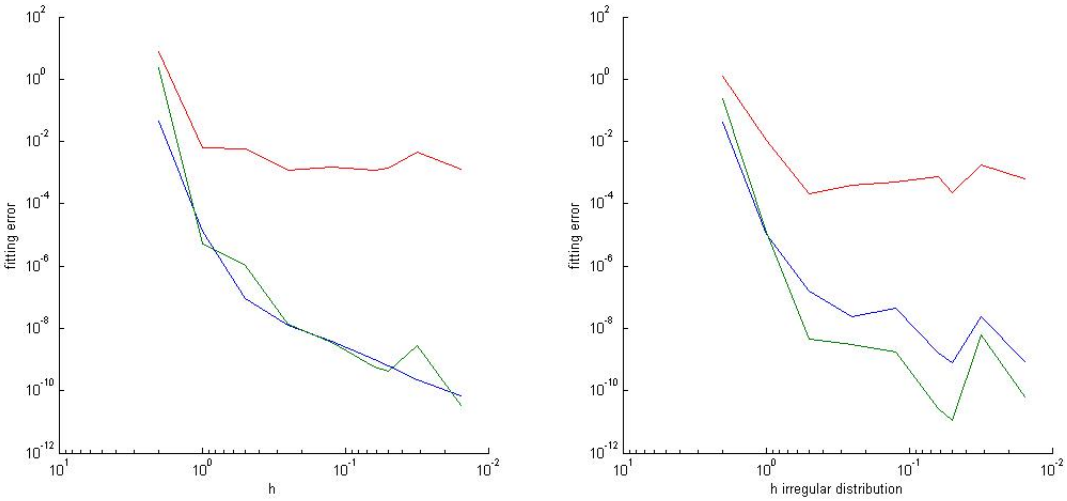


Figure 4.37: Comparison of the fitting error's convergence of RPIM-EXP for regular (left) and irregular (right) nodal distribution. $\alpha_c = 0.03$.

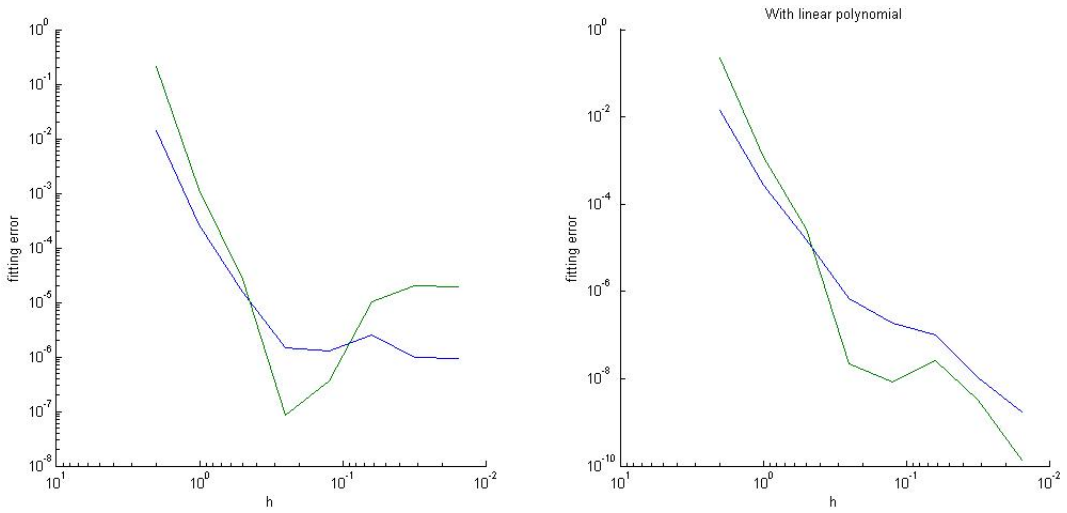


Figure 4.38: Convergence of fitting error for RPIM-WEN C6 without polynomial term (left) and with polynomial term (right). irregular distribution of nodes.

Figure 4.39 demonstrates that adding polynomial term for CSRBF, has a negligible effect on the conidtioning of the moment matrix.

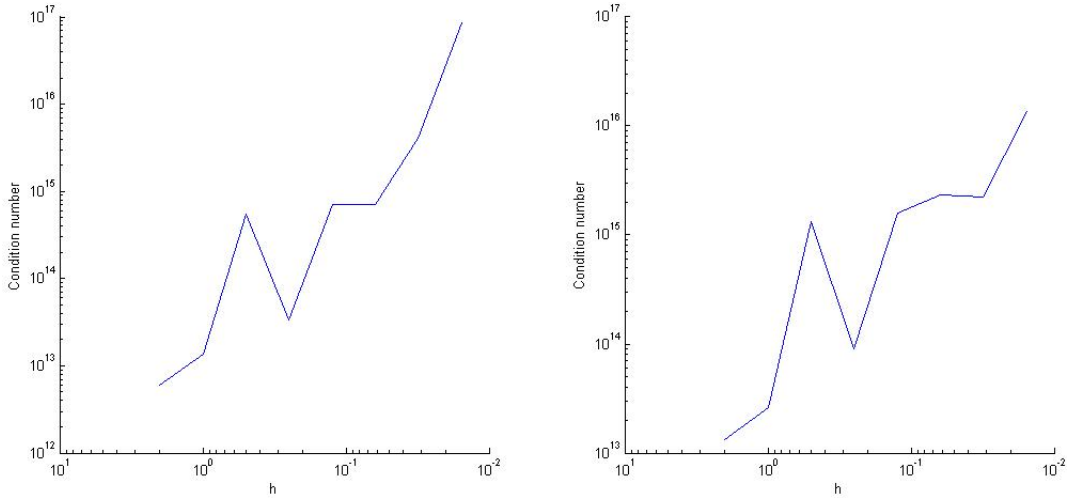


Figure 4.39: Influence of refinement h on the condition number of the moment matrix for RPIM-WEN C6 without polynomial term (left) and with polynomial term (right). irregular distribution of nodes.

Through the different tests, it appears that adding a linear polynomial to the basis for a majority of problems provides a slightly better accuracy and improved stability for conventional RBF. Similar results are obtained with other CSRBF.

4.3.2 Fitting accuracy of the MLS interpolant

In order to study and assess the impact of different parameters such as the support domain or the basis's size on the MLS's interpolation quality, we consider a square domain $\Omega\ (x, y) \in [0, 10] \times [0, 10]$ discretised by a set of regular distribution of field nodes. We first use a 31×31 regular distribution of nodes. A set of 10×10 sampling points is used as interpolation points and the linear basis (m=3) with the cubic spline is used. In the following tests, we use the same error norm defined by equation 4.87 and 2D surface function given by equation 4.88.
Figure 4.40 (left) shows the impact of the support domain size α_s on the error norm for the function and its partial derivatives up to the $3^r d$ order.

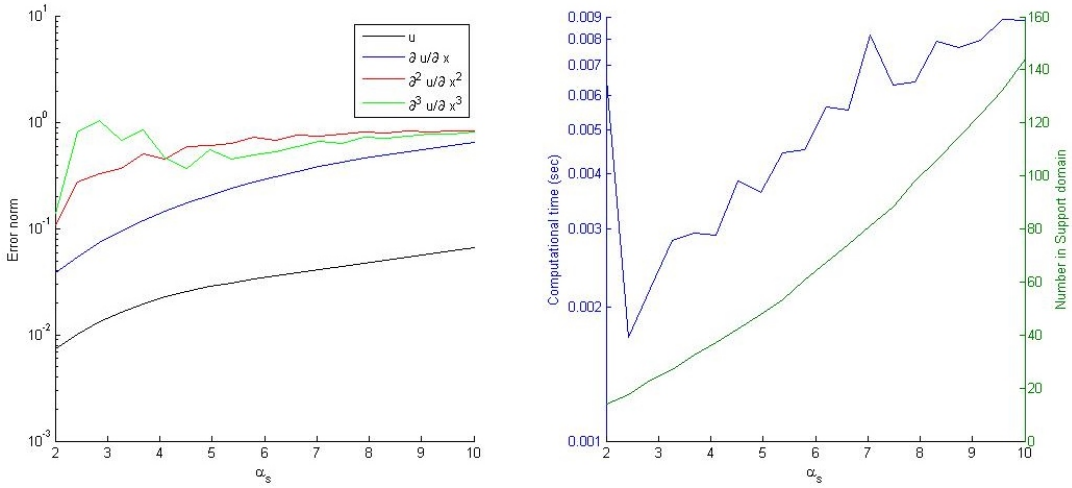


Figure 4.40: Impact of the support domain size on MLS error fitting and computational time for different support domain

Figure 4.40 (right) also highlights the impact of α_s on the computational time (blue line). Green line represents the relation between the size α_s and the number of node included in the support domain.

It appears that compact support with dimensionless parameter $\alpha_s \in [2, 5]$ offers the best fitting accuracy while being the least expensive computational-wise. According to the test, the optimal number of field nodes within a support domain for a linear basis appears to be in the range of 15-40. The condition number of the moment matrix A remains small, independent of the support domain size. In practice, the increase of the size of the support domain does not improve the interpolation accuracy. A similar conclusion can made for irregular distribution of nodes as shown on figure 4.41.

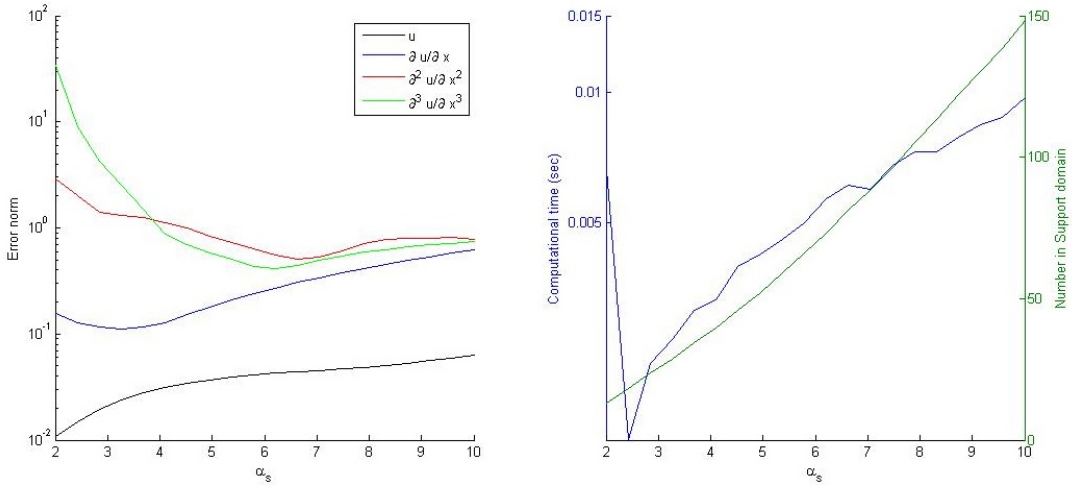
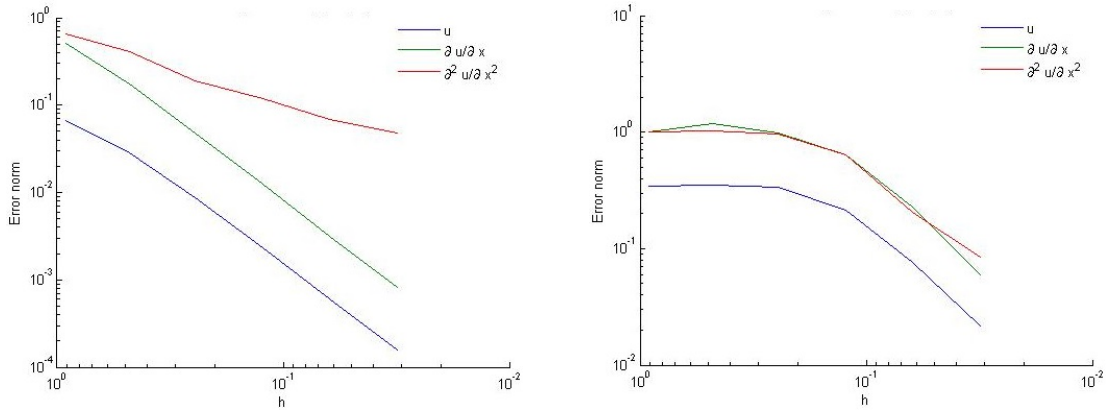


Figure 4.41: Impact of the Support domain size on MLS error fitting for irregular distribution of node

Figure 4.42 shows the h-convergence assessment of the MLS interpolant for two value of k. For the case where $k = \pi/6$, the MLS interpolant possesses a logarithm type of convergence. However in the case where $k = 3\pi$ where the surface is much more oscillatory, the convergence for the same parameter value of α_s, does not achieve the logarithmic convergence rate.

A similar fitting accuracy is only obtained at a much higher density of field nodes than for the



(a) k=$\pi/6$ (b) k=3π

Figure 4.42: Convergence of error norm for surface interpolation

smoother surface interpolation.

In figures 4.40 - 4.42, the interpolation of the second derivative is less accurate than the first order derivative and function. This is mainly due to the choice of the basis along with the size of the support domain. For higer derivative of the basis it produces a rank deficiency in the system 4.31 generates a badly conditioned A matrix.

The effect of different sizes of monomial basis $p(x)$ on the MLS SF is observed on figure 4.43. As one can see they are smooth for all orders m. It is important to note that increasing the order of

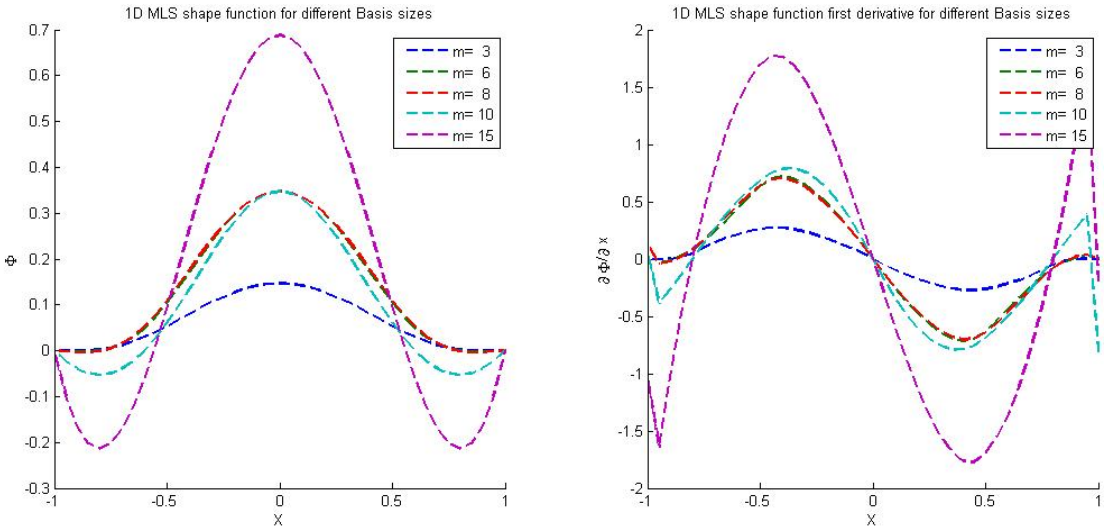


Figure 4.43: 1D MLS shape function for different monomial basis size.

the basis does not provide a smoother fitting as shown on figure 4.44. This characteristic results from the weight function smoothness, but it improves convergence rate.

However, the augmentation of the basis increases the size of the Moment matrix bringing major drawbacks. There is a need to include more nodes in the support domain in order to fulfil the rank condition for the moment matrix, and allow its inversion so as to calculate an accurate SF. Increasing the order of the MLS basis, and thus the size of the moment matrix, leads to a drastic increase in the computational cost of the shape function generation. This is highlighted in section 6.1 p.131 through a 2D heat conduction benchmark problem.

The MLS shape function can reproduce any function present in its basis exactly. This feature is particularly suitable for problems with known singularity such as crack problems where singular

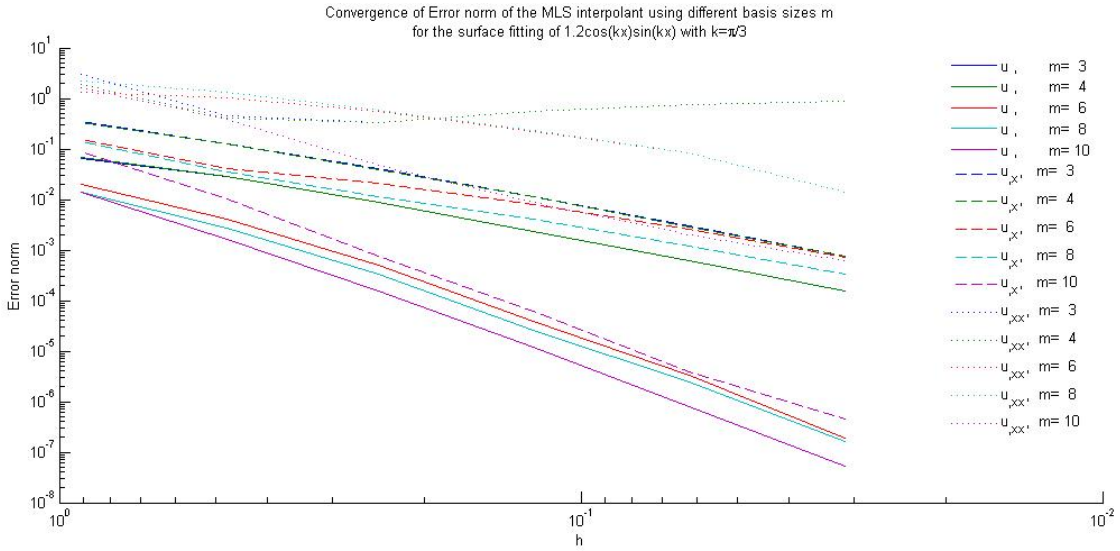


Figure 4.44: Impact of the size of the basis on MLS error fitting convergence

stress field appears near the tip of the crack [31, 32]. However, enriching the basis with functions often generates a not so well-conditioned moment matrix A.

4.4 Conclusion

In this chapter we have introduced the concept of meshfree method and reviewed the major techniques used in building meshfree SFs. Robust generation of SFs on a scattered set of nodes is one of the most challenging aspects of meshfree methods. Through the review of the different techniques, the building process of MF SFs is revealed to be more much more complicated and computationally expensive compared to building FEM polynomial SFs using Serendipity element. A difficulty in manipulating meshfree methods is the choice of parameters to tune the approximation. In this chapter, we characterised the meshfree MLS and RPIM SFs through the study of their capibility for surface fitting problems. It allowed us to define sets of parameters that can be used for practical implementation.

Overall, these SFs work very well for fitting problems. Usually meshfree SFs exhibits higher rate of convergence compared to the standard FEM or FDM. However special attention needs to be given to the moment matrix in the building process. This matrix can become badly conditioned when parameters are not optimally defined, or when non-optimal coverage is used to build the interpolation. This results in the need for special treatment to solve the linear system or for the inversion of the matrix. Ultimately it will have a negative impact on the computational cost associated with the building of those functions for practical implementation. Nonetheless, MLS and RPIM SFs are stable and have been chosen as staple methods for our application.

Meshfree methods can be either implemented using weak formulation such as the one use in FEM or collocation methods. In the next chapter we introduce the weak formulation based methods and review their implementation procedure.

Chapter 5 Meshfree methods based on weak formulation

Following the introduction and review of meshless methods to build shape functions, this chapter focuses on the implementation of the weak formulation based methods and the review of critical aspects of the procedures. This chapter also seeks to characterise the behaviour of MLS and RPIM meshfree interpolants using benchmark problems. The goal is to assess the key features that produce optimum results in terms of accuracy and computational cost for their future use in practical problems.

5.1 Meshfree based on global weak form

In 1968, Shepard [259] presented a meshless interpolation for irregularly spaced data points. After the introduction of the MLS method by Lancaster and Salkauskas in 1981 [146], Nayroles et. al. [201], employed a local form of this approximation for the numerical solution of some PDE's. Their method named Diffuse Element Method (DEM) uses a truncated form for derivatives of the shape function. In 1994 Belytschko et. al. [280] generalized the DEM and introduced the element free Galerkin (EFG) method. Kansa was the first to use the radial function (RBF) as an interpolant within a collocation type formulation [132, 133] followed by Schaback [246] and Wendland [295] who introduced the RBF in the same formulation as the EFG. The flowchart 5.1 p.94 describes the global EFG procedure.

5.1.1 Element Free Galerkin method (EFG)

To illustrate the EFG method we consider the two-dimensional strong form (in matrix form) of the elasto-static problem over the domain Ω along with its boundary Γ.

$$L^T\sigma + b = 0 \quad in\ \Omega \tag{5.1}$$

$$\sigma n = \bar{t} \quad on \quad \Gamma_t \tag{5.2}$$

$$u = \bar{u} \quad on \quad \Gamma_u \tag{5.3}$$

With u being the displacement vector $u^T = \{u, v\}$ and L the 2D matrix of differential operator

$$L = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \tag{5.4}$$

σ^T is the stress vector $\{\sigma_{xx}\sigma_{yy}\sigma_{xy}\}$, $b^T = \{b_x, b_y\}$ the body force vector. $\bar{t}$ and $\bar{u}$ are the prescribed traction and prescribed displacement respectively. n is the unit outward normal to $\Gamma \equiv \Gamma_t \cup \Gamma_u$ such that $\Gamma_t \cap \Gamma_u = 0$. Using the constitutive equations, we can express the stress-

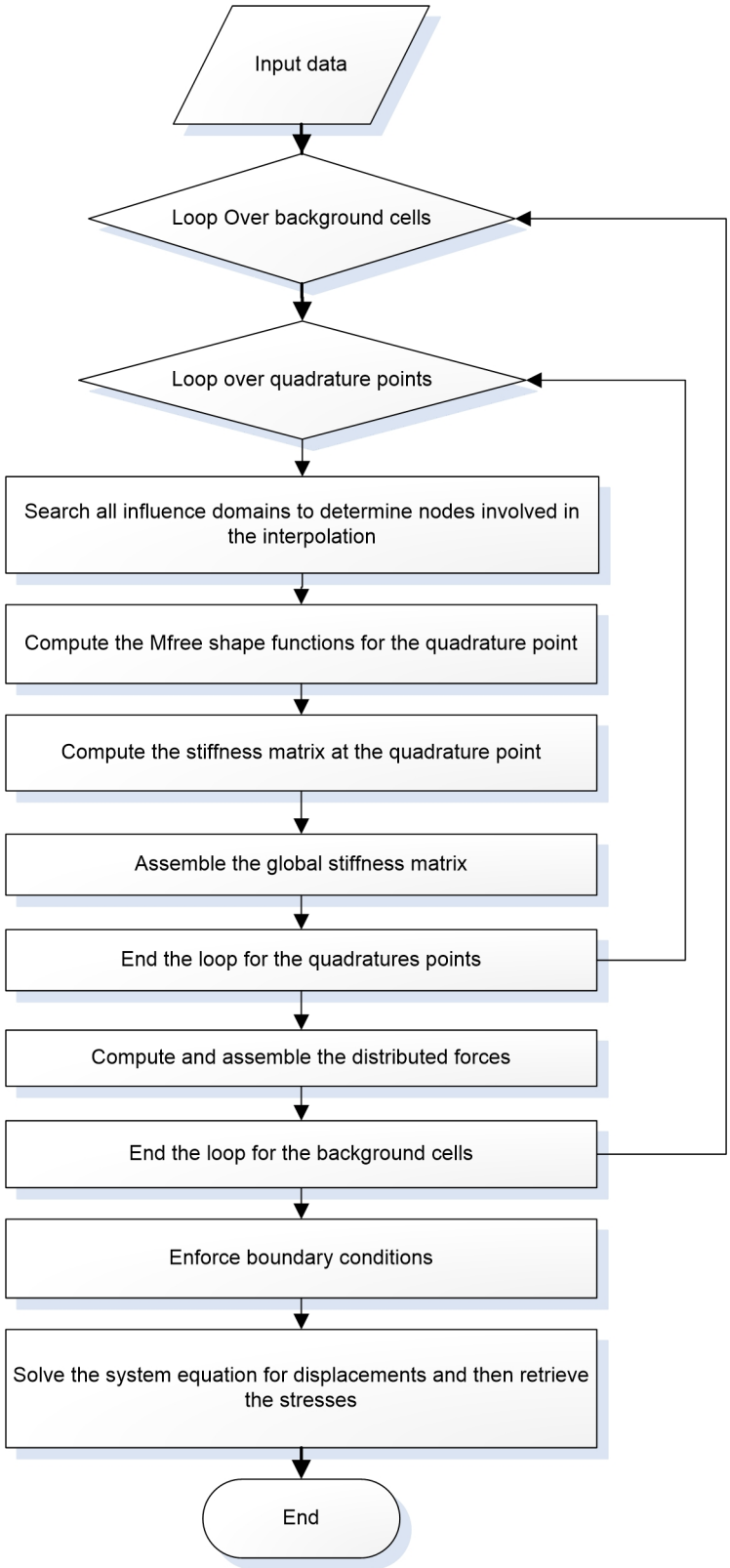


Figure 5.1: EFG flowchart

strain relationship

$$\sigma = D\epsilon \tag{5.5}$$

with the $\sigma = \{\sigma_{xx}\sigma_{yy}\tau_{xy}\}$ being the stress vector components and $\epsilon = \{\epsilon_{xx}\epsilon_{yy}\epsilon_{xy}\}$ the strain vector components. D is the matrix of material elastic constants which is derived from empirical and experimental measurements. In the case of a plane stress problem such as the linear 2D elastic beam problem, we have

$$D = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{5.6}$$

Now using the strain-displacement relationship given by

$$\epsilon = Lu \tag{5.7}$$

We have

$$\sigma = DLu \tag{5.8}$$

The first step involves taking the weak formulation using either the Lagrangian principle of least action [55, 59, 91, 95, 269], variational principle, the energy principle [95] or the weight residual method (WRM) (see appendix A p. 204) to obtain the Galerkin weak formulation. Using the minimum total potential energy principle (see section B.2) and equation 5.8, the following standard weak formulation is derived

$$\int_\Omega (L\delta u)^T (DLu) d\Omega - \int_\Omega \delta u^T b d\Omega - \int_{\Gamma_t} \delta u^T \bar{t} d\Gamma = 0 \tag{5.9}$$

From this point onwards, the major difference between FEM and EFG occurs. The problem domain is discretised using a set of field nodes in order to approximate the field variable, in this case the displacement, using any meshfree SF presented in section 4.2. Using MF SFs gives the following approximation for the displacement at any point of interest x_I

$$u^h(x_I) = \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \cdots & \phi_n & 0 \\ 0 & \phi_1 & 0 & \phi_2 & \cdots & 0 & \phi_n \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_n \\ v_n \end{Bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} \phi_i & 0 \\ 0 & \phi_i \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} = \Phi u \tag{5.10}$$

n corresponds to the number of nodes with the support domain of a point of interest x_I. Φ is the vector containing the shape function associated with each fields nodes. u is the vector collecting the nodal displacement for the field node included in the support domains of the point of interest x_I. The field nodes, used to build the discretised system of equation, are uniquely numbered from 1 to N. This is important for the assembly procedure of local nodal matrix into the global system

matrix. The variation of the displacement field can be written

$$\delta u^h(x_I) = \sum_{i=1}^{n} \Phi_i \delta u_i = \Phi \delta u \tag{5.11}$$

Using the strain-displacement relation

$$\epsilon = Lu^h = \sum_{i=1}^{n} \Phi_i u_i = L\Phi u = Bu \tag{5.12}$$

B is the strain matrix for 2D problem at the node x_I and is defined

$$B = \begin{pmatrix} \frac{\partial \phi_1}{\partial x} & 0 & \cdots & \frac{\partial \phi_n}{\partial x} & 0 \\ 0 & \frac{\partial \phi_1}{\partial x} & 0 & \cdots & \frac{\partial \phi_n}{\partial y} \\ \frac{\partial \phi_1}{\partial y} & \frac{\partial \phi_1}{\partial x} & \cdots & \frac{\partial \phi_n}{\partial y} & \frac{\partial \phi_n}{\partial x} \end{pmatrix} \tag{5.13}$$

Thus

$$L\delta u^h(x_I) = L\Phi \delta u = L\sum_{i=1}^{n} \Phi_i \delta u_i = B\delta u \tag{5.14}$$

For 2D elastic solids, using the Hooke's law or constitutive equation, we can express the stress vector

$$\sigma = D\epsilon = DBu = \sum_{i=1}^{n} DB_i u_i \tag{5.15}$$

Using equation 5.15 and 5.12 we can express the stress vector

$$\sigma = DLu^h = DL\Phi u^h = DBu \tag{5.16}$$

Substituting equation 5.16 and equation 5.14 into equation5.9

$$\int_\Omega \sum_{i=1}^{n} (B_i \delta u_i)^T \sum_{j=1}^{n} DB_j u_j d\Omega - \int_\Omega (\sum_{i=1}^{n} \Phi_i \delta u_i)^T b d\Omega = \int_{\Gamma_t} (\sum_{i=1}^{n} \Phi_i \delta u_i)^T \bar{t} d\Gamma \tag{5.17}$$

$$\int_\Omega \sum_{i=1}^{n} \sum_{j=1}^{n} \delta u_i^T (B_i^T D B_j) u_j d\Omega - \int_\Omega (\sum_{i=1}^{n} \Phi_i \delta u_i)^T b d\Omega = \int_{\Gamma_t} (\sum_{i=1}^{n} \Phi_i \delta u_i)^T \bar{t} d\Gamma \tag{5.18}$$

Moving from local node numbering (i,j) in the support domain to a global numbering system (I,J) through the whole domain Ω allows us to move the integral within the summation.

$$\sum_{I=1}^{N} \sum_{J=1}^{N} \delta u_I^T \int_\Omega (B_I^T D B_J) d\Omega u_J - \sum_{I=1}^{N} \delta u_I^T \int_\Omega \Phi_I^T b d\Omega = \sum_{I=1}^{N} \delta u_I^T \int_{\Gamma_t} \Phi_I^T \bar{t} d\Gamma \tag{5.19}$$

From equation 5.19, we can observe the assembly process resulting from the form of the equation. The first term of the LHS can be written

$$\sum_{I=1}^{N} \sum_{J=1}^{N} \delta u_I^T \int_\Omega (B_I^T D B_J) d\Omega u_J = \sum_{I=1}^{N} \sum_{J=1}^{N} \delta u_I^T K_{IJ} d\Omega u_J = \delta U^T K U \tag{5.20}$$

where K_{IJ} is the nodal stiffness matrix. Here we can see the similarity in the FEM assembly process. The size of the stiffness matrix is $kN \times kN$. k is the degree of freedom. Similarly, the

second term of the LHS can be written

$$\sum_{I=1}^{N} \delta u_I^T \int_{\Omega} \Phi_I^T b d\Omega = \sum_{I=1}^{N} \delta u_I^T F_I^b = \delta U^T F^b \tag{5.21}$$

Where F_I^b is the nodal body force vector. For the RHS term we have

$$\sum_{I=1}^{N} \delta u_I^T \int_{\Gamma_t} \Phi_I^T \bar{t} d\Gamma = \sum_{I=1}^{N} \delta u_I^T F_I^t d\Gamma = \delta U^T F^t \tag{5.22}$$

With F_I^t the nodal traction force vector. This weak formulation is defined over the whole domain Ω. In order to evaluate the integral in equation 5.9, Belytschko [280] relied first on a background of non-overlapping cells. This topic is extensively studied in literature and constitutes an active area of research with focus areas such as stabilization procedures. Similarly a set of curves is used to evaluate the natural boundary integral equations. Using a Gauss quadrature scheme over the background cell for support allows us to rewrite the nodal stiffness matrix in equation5.20

$$K_{IJ} = \int_{\Omega} (B_I^T D B_J) d\Omega = \sum_{k=1}^{n_c} \sum_{l=1}^{n_g} w_l B_I^T(x_{ql}) D B_J(x_{ql}) \left|J_{lk}\right| \tag{5.23}$$

n_g is the total number of Gauss points used in one cell. n_c is the total number of cells. The nodal stiffness matrix is evaluated at each quadrature point x_{ql} for each cell of the background grid. $|J_{lk}|$ is the Jacobian matrix for the area integration of the cell k. w_l corresponds to the weight associated with the Gauss point x_{ql}. Similarly for the nodal vector components we can write the nodal body force vector

$$F_I^b = \sum_{k=1}^{n_c} \sum_{l=1}^{n_g} w_l \delta(x_{ql}) b \left|J_{lk}\right| \tag{5.24}$$

For the nodal traction, the integral is carried over the Gauss curves

$$F_I^t = \int_{\Gamma_t} \Phi_I^T \bar{t} d\Gamma = \sum_{m=1}^{n_c t} \sum_{l=1}^{n_g t} w_l \Phi_I^T(x_{ql}) \bar{t}(x_{ql}) \left|J_{lm}^{\Gamma}\right| \tag{5.25}$$

$\left|J_{lm}^{\Gamma}\right|$ is the curve Jacobian. $n_c t$ is the number of sub-curve boundary used to discretised the boundary domainΓ. $n_g t$ is number to Gauss point for the integration curve w_l the weight associated with the quadrature point x_{ql}. Substituting equations 5.20 , 5.21 and 5.22 into 5.19, gives

$$\delta U^T K U - \delta U^T F^b - \delta U^T F^t = 0 \tag{5.26}$$

thus

$$\delta U^T (K U - F^b - F^t) = 0 \tag{5.27}$$

Now δU being admissible but arbitrary, equation 5.27 holds for any value. Thus we obtain the final system of linear equations

$$K U = F \quad with \quad F = F^b + F^t \tag{5.28}$$

F is the global force vector like in the FEM final system of equation. After enforcing the essential boundary conditions, solving the system 5.28 will give the nodal displacements.
When the MLS is used as approximant in equation 5.19, due to the lack of the delta property, the essential boundaries condition cannot be applied by a direct method. In its original formulation Belytschko used Lagrange multipliers [280] but other methods can be employed to enforce these essential boundaries conditions (see section 5.1.3). In the case of RPIM, direct methods can be used.
The difference with FEM lies in the nodal construction. In FEM, SFs are built over the mesh element and all the Gauss points in one element share the same SF and element nodes to build the approximation. In EFG each quadrature point $x_q l$ possesses its own support domain on which it builds the shape function. Thus the matrix B varies from one quadrature point to another.
The assembly process is based on the support domain of the quadrature point. Simply when the nodes I and J are not in the support domain of the quadrature point, the nodal contribution vanishes. The compactness of the weight function and the support domain ensure that only local points are used in the building of the nodal matrix. These features are important because they will produce sparse global stiffness matrix K. In equation 5.20, the matrix D given by equation 5.6 is symmetric. Thus the nodal stiffness matrix K_{IJ} and global K will also be symmetric. Finally depending on the field node numbering, the matrix can also be banded.

5.1.2 Compatibility

When using the global weak formulation such as the energy principle to derive the weak formulation of the problem, the approximated field function has to be continuous over the whole problem domain Ω. This is referred as the compatibility criterion. Meshfree methods based on a global weak formulation require a set of background cells for the numerical integration. Because global weak formulation requires the system equation in the global integral form to be satisfied over the entire problem domain Ω, incompatibility through the domain appears in between two support domains when non compatible shape functions are used. Gaps or overlaps of the support domain can affect the system energy and break the balance of energy principle.
Compatibility can be imposed directly through a constrained Galerkin weak form on the incompatible interface. For instance, for a 2D elastic problem, we consider the curve Γ_c on which incompatibility exists. As described in [162] the compatibility condition on Γ_c can be enforced using the following constrained weak form

$$\int_\Omega (L\delta u)^T (DLu) d\Omega - \int_\Omega \delta u^T b d\Omega - \int_{\Gamma_t} \delta u^T \bar{t} d\Gamma + \int_{\Gamma_c} \delta(u^+ - u^-)\alpha(u^+ - u^-) d\Gamma \quad (5.29)$$

α is a matrix of penalty factors and u^+ and u^- are the displacements on the two sides of the incompatible interface Γ_c. For instance in the case where RPIM is used to build SF, the interpolation is not always compatible over the global domainΩ. The regular global weak form based MF using RPIM SF is often referred to as non-conforming RPIM (NCRPIM) whereas the conforming RPIM using the constrained weak form is termed CRPIM. CRPIM is more accurate than NCRPIM and works fine for many applications, but is computationally more expensive. However, similarly as in FEM with incompatible SFs the solutions can still converge and in some cases, even faster than with compatible shape functions.

5.1.3 Boundaries conditions

In the global weak form based meshfree methods, the implementation of Dirichlet boundary conditions is a main concern because not all the methods used to create the SF provides the Kronecker delta property. Therefore, imposing boundary conditions is not as straightforward as in FEM with the direct imposition method. A lot of research has been carried out to overcome this drawback. There is class of techniques:

- Methods based on modification of the weak formulation, such as the Lagrange multiplier method [30], the penalty method [306] and Nitsche's method [14, 103].
- Methods using direct modification of the SFs [97, 123, 141, 223].

Methods using a modified weak formulation allow the use of trial functions that do not vanish at the essential boundary. The Lagrange multiplier method is one of the most widely used because of its straightforward implementation with all kinds of problems. This method introduces a new unknown function, the Lagrange multiplier. The interpolation space for the Lagrange multiplier must be carefully selected. It has to be rich enough in order to obtain an acceptable solution, but the resulting system of equations will turn out to be singular if the number of degrees of freedom for the discretisation is too large [109].
On the other hand, the penalty method and Nitsche's method require only the choice of one scalar parameter. In the penalty method, large values of the parameter must be used in order to impose the essential boundary condition properly. In practice, it can lead to ill-conditioned systems of equations, reducing the applicability of this method. On the contrary, Nitsche's method does not suffer ill-conditioning. However, the implementation of Nitsche's method is not as simple as for the Lagrange multiplier method or the penalty method, in the sense that the modification of the weak form is different for each different problem.

Methods based on the modification of the shape functions have also been studied. Chen developed a transformation method that expresses the meshfree unknowns as a linear combination of nodal unknowns [56]. This method allows the definition of shape functions that verify the delta property. For RPKM, Gosz [97] introduced an extension of the dilation parameter at each particle and then the meshfree shape functions is forced to verify the Kronecker delta property at the boundary. D'Alembert principle has also been used for meshfree methods [90] for the imposition of all kind of linear constraints. Orthogonality of the constraint matrix is assumed in order to express the unknowns as a linear combination of a set of generalized unknowns. Essential boundary conditions are directly imposed on these generalized unknowns. Transformation methods are usually employed in transient, or evolution problems.

Another solution is to modify the meshfree shape functions to couple with a finite element interpolation near the essential boundaries to directly impose prescribed values [123, 124, 141, 223, 289]. The coupling must be consistent with the local approximation orders of the local spaces. Otherwise it impacts the approximation quality of the overall method. Belytschko [223] proposed a coupled interpolation where both finite elements and particles have an influence. It requires the substitution of finite element nodes by particles and the definition of ramp functions. There is a transition region of the size of one finite element and the interpolation is linear. Huerta presented a unified and general formulation for a continuous blending [123].

Wagner [289] also proposed the bridging scale method to mix a meshfree approximation with any other interpolation space near the essential boundary. However the meshfree shape functions vanish at the boundary nodes but not at the whole essential boundary. The rate of convergence is decreased because the test functions do not cancel along the Dirichlet boundary. A modified weak form must be used to impose the boundary condition correctly. A detailed comparison between the continuous blending method and the bridging scale method can be found in [124].
Coupling meshfree method with a mesh-based method removes the meshfree character of the original method just for the sake of the implementation of Dirichlet boundary conditions which results in a loss of generality and freedom of the overall method.

5.1.3.1 Lagrange multipliers

The Lagrange multiplier method is a general approach towards the solution of constrained minimisation problems which is also used in the FEM [13] and wavelet [142] context to implement essential boundary conditions. The choice of the multipliers space for optimal converge is addressed in [219, 220].
In Lagrange multipliers methods the following essential boundary condition 5.3 functional is added to the Lagrangian 5.9

$$\int_{\Gamma_u} \lambda^T (u - \bar{u}) d\Gamma \tag{5.30}$$

The functional related to the essential boundary is then

$$\delta \int_{\Gamma_u} \lambda^T (u - \bar{u}) d\Gamma = \int_{\Gamma_u} \delta\lambda^T (u - \bar{u}) d\Gamma + \int_{\Gamma_u} \lambda^T \delta u d\Gamma \tag{5.31}$$

Thus the weak formulation 5.9 incorporating the essential boundary conditions through lagrange multipliers is given by

$$\int_\Omega (L\delta u)^T (DLu) d\Omega - \int_\Omega \delta u^T b d\Omega - \int_{\Gamma_t} \delta u^T \bar{t} d\Gamma - \int_{\Gamma_u} \delta\lambda^T (u - \bar{u}) d\Gamma - \int_{\Gamma_u} \lambda^T \delta u d\Gamma = 0 \tag{5.32}$$

In order to impose the $(u - \bar{u}) = 0$, lagrange multipliers λ are interpolated using their nodal values

$$\lambda^h = \sum_{i=1}^{n_\lambda} \begin{bmatrix} L_i & 0 \\ 0 & L_i \end{bmatrix} \begin{Bmatrix} \lambda_{u_i} \\ \lambda_{v_i} \end{Bmatrix} = \sum_{i=1}^{n_\lambda} L_i \lambda_i = L(s)\lambda \tag{5.33}$$

L_i is the shape function for the i^{th} node of the essential boundary. n_λ is the number of nodes used for the interpolation. The term s defines the arc length along the essential boundary Γ_u. λ is the vector containing the nodal Lagrange multipliers for each field nodes on the boundary. The L_i are the conventional Lagrange SFs used for Lagrangian element in FEM. Lagrange interpolant of order n is given as

$$L_i^n(s) = \frac{(s - s_0)(s - s_1) \cdots (s - s_{i-1})(s - s_{i+1}) \cdots (s - s_n)}{(s_i - s_0)(s_i - s_1) \cdots (s_i - s_{i-1})(s_i - s_{i+1}) \cdots (s_i - s_n)} \tag{5.34}$$

In our case we use the linear first order Lagrange interpolant. Thus for the two verticess_0 and s_1

of a segment sub-boundary ofΓ_u we have

$$L_0(s) = \frac{(s - s_1)}{(s_0 - s_1)}; \quad L_1(s) = \frac{(s - s_0)}{(s_1 - s_0)} \tag{5.35}$$

Using equation5.33 and the fact that $\delta\lambda^h = L\delta\lambda$, we can substitute into equation 5.31 to obtain the first term of the RHS

$$\int_{\Gamma_u} \delta\lambda^T(u - \bar{u})d\Gamma = \int_{\Gamma_u} \delta \left(\sum_{I=1}^{n_{\lambda_t}} L_I\lambda_I\right)^T \sum_{J=1}^{n} \Phi_J u_J d\Gamma - \int_{\Gamma_u} \delta \left(\sum_{I=1}^{n_{\lambda_t}} L_I\lambda_I\right)^T \bar{u}d\Gamma \tag{5.36}$$

n_{Λ_t} is the total number of nodes on the Γ_u. Permuting summation and integral gives

$$\int_{\Gamma_u} \delta\lambda^T(u - \bar{u})d\Gamma = \sum_{I=1}^{n_{\lambda_t}} \sum_{J=1}^{n} \delta\lambda_I^T \int_{\Gamma_u} L_I\Phi_J d\Gamma u_J - \sum_{I=1}^{n_{\lambda_t}} \delta\lambda_I^T \int_{\Gamma_u} L_I^T \bar{u}d\Gamma \tag{5.37}$$

We write in matrix form

$$G_{IJ}^T = -\int_{\Gamma_u} L_I\Phi_J d\Gamma; \quad Q_I = -\int_{\Gamma_u} L_I^T\bar{u}d\Gamma \tag{5.38}$$

Thus we obtain the system

$$\int_{\Gamma_u} \delta\lambda^T(u - \bar{u})d\Gamma = -\delta\Lambda G^T U + Q \tag{5.39}$$

Λ is the vector for the nodal Lagrange multipliers of each field nodes on Γ_u. Similarly, the second term of the RHS of equation 5.31 gives

$$\int_{\Gamma_u} \lambda^T\delta u d\Gamma = \int_{\Gamma_u} \delta u^T\lambda d\Gamma = \int_{\Gamma_u} \left(\sum_{I=1}^{n} \Phi_I\delta u_I\right)^T \sum_{J=1}^{n_{\lambda_t}} L_J\lambda_J d\Gamma \tag{5.40}$$

Once again using linearity and permuting the integration and the sum gives

$$\int_{\Gamma_u} \delta u^T\lambda d\Gamma = \sum_{I=1}^{n} \sum_{J=1}^{n_{\lambda_t}} \delta u_I^T \int_{\Gamma_u} \Phi_I^T L_J d\Gamma\lambda_J = \sum_{I=1}^{n_{\lambda_t}} \sum_{J=1}^{n} \delta u_I^T \left[-G_{IJ}\right]\lambda_J = -\delta U^T G\Lambda \tag{5.41}$$

Integration of the nodal matrix 5.39 is performed using Gauss quadrature. Substituting 5.39 and 5.41 into 5.32 gives

$$\delta U^T \left[KU - F + G\Lambda\right] + \delta\Lambda^T \left[G^T U + Q\right]) = 0 \tag{5.42}$$

Because equation 5.42 holds for any $\delta\Lambda^T$ and δU we obtain the following system of equation

$$\begin{bmatrix} K & G \\ G^T & 0 \end{bmatrix} \begin{Bmatrix} U \\ \Lambda \end{Bmatrix} = \begin{Bmatrix} F \\ Q \end{Bmatrix} \tag{5.43}$$

The final system of linear equation is augmented. Depending on the number of nodes along the boundary Γ_u, the number of added Λ variables in the linear system can influence the overall computational cost. The final system matrix in 5.43 is non positive definite and not banded. However the matrix is still symmetric.

5.1.3.2 Penalty method

The penalty method introduces an additional surface term in the variational formulation to enforce the boundary conditions. This penalty term may change the properties of the functional. It only requires multiplication of the stiffness matrix term by the penalty factor and the modification of the corresponding force vector term. This results in a minimal cost on a computational level. An important feature with the penalty method is that it conserves the dimension, symmetry and positive definite properties of the system stiffness matrix.
One major issue is to determine the optimum penalty coefficient. Based on practice for both FEM and EFG, the penalty coefficient α can be defined as

$$\alpha = 10^3 - 10^{10} \times (K_{ii})_{max} \tag{5.44}$$

With $K_{ii_{max}}$ being the maximum diagonal element of the global stiffness matrix. The penalty or perturbation approaches are very general concepts for the implementation of constraints in a variational problem. The penalty method only ensures the essential boundaries conditions approximately and decreases the rate of convergence. This will impact the overall approximation quality of the meshfree method due to an inappropriate treatment of boundary conditions. The effect is usually minor and hard to estimate.

5.1.4 Integration techniques

A very sensitive process in weak-formulation based meshfree methods is the integration procedure. In Galerkin formulation, the PDE's are transformed into a set of algebraic linear equations where continuous integrals need to be approximated by a numerical quadrature. As already mentioned, the computational efficiency of Galerkin methods is strongly related to the accuracy of the Integration scheme employed [7, 26, 29, 159]. Different approaches have been developed for the integration of the weak formulation of PDE's equation in meshfree methods.

5.1.4.1 Cell quadrature

This method requires a regular background mesh to perform the regular Gauss quadrature. This is the technique originally used in EFG method. A study of this method of integration was can be found in [74, 102]. The integration is operated in a similar way as it is done with the FEM technique over the elements. There are two major issues with this technique:

- The local support domain used to select the nodes participating in the construction of the shape functions usually does not align with the integrations domains [74]. This is not a problem with the standard FEM as the integration is led over the element which is also used to build the polynomial shape functions.

- Techniques such as the Gauss quadrature are methods suited to integrate polynomial functions such as shapes functions used in FEM exactly. However MF SFs are complex and rational non polynomial function, which results in errors in accuracy and convergence when using the regular Gauss quadrature.

These two shortcomings of the Galerkin-weak-formulation meshfree methods lead to the requirement of more quadrature points than is required with the standard FEM. Hence, for meshfree

methods the procedure is computationally more expensive than in FEM.

5.1.4.2 Stress point integration method

This method was originally developed by Dyka [77] to stabilize the original SPH method and correct tensile instability. The technique uses additional slave particles called stress points added between the original particles which serve as additional quadrature points. The method shows good convergence for uniform nodes distribution but has problems for non-uniform nodes. Randles and Libersky [230] extended the method and only used the stress points as quadrature points. Unfortunately, it has been reported that the stress points technique does not solve tensile instability problems [74]. In many cases they restore the positive definiteness of the linear equations and eliminate the instabilities due to rank deficiency. Rabczuk [228] presented a new form of stress-point integration for multi-dimensions that employs both the particles and stress points as quadrature points. This technique is reported to provide better stability properties compared to the original nodal integration technique. Fries and Belytschko studied the convergence of the nodal integration [86] based on stress point integration and showed that for regular nodal distribution, the rate of convergence is good. However, for non-uniform nodal arrangements, stress-point integration is associated with a mild instability which can result in poor convergence.
Stabilization schemes based on least-square (LSS) [26, 86] and Taylor series expansion (TEBS) [164] and the finite increment gradient (FIG) stabilization [41] have been developed to restore good rate of convergence. Results show that LSS and TEBS have successfully improved the convergence and stability properties of stress-point integration where the FIG fails to. Moreover, the superiority of stabilized stress-point integrations over the stabilized nodal integrations has been demonstrated.

5.1.4.3 Nodal integration

The Nodal integration scheme evaluates the integrals of the weak form only at the nodes. This method is attractive due to the relatively high computational efficiency and the absence of a background cell structure or background mesh for the Gauss quadrature. The quadrature is directly expressed

$$\int_{\Omega} u(x)d\Omega = \sum_{i=1}^{N} u(x_i)\Delta\Omega_i \tag{5.45}$$

With N being the number of particles in the particle of interest support.
This integration method introduces stability problems that are also encountered in collocation methods and particle methods such as SPH. The under-integration of the weak form introduces spurious singular modes. Beissel [26] developed the stabilized nodal integration method by adding the residual of the equilibrium equation to the energy function used by the EFG. This quadrature method is the fastest but results in some instabilities when integrating high order differential equations (k>2) [27].

5.1.4.4 Stabilized conforming nodal integration(SNCI)

The stabilized conforming nodal integration (SNCI) have been developed by Chen et al. to solve the instability problem [11, 57, 58, 226, 228, 275] and improve gradient continuity. This technique

is based on a strain smoothing procedure over cell surrounding the nodes. The strain smoothing or gradient smoothing is written

$$\tilde{\nabla} u_i^h = \frac{1}{A_k} \int_{\Omega_k} \nabla u_i^h d\Omega \tag{5.46}$$

With A_k being the area of smoothing domain for node k. Ω_k is a nodal representative domain and $\tilde{\nabla}$ is the smoothed gradient operator. Different strategies can be used to build the nodal smoothing domain Ω_k

- The first approach is to use Voronoi diagram cells. The Voronoi diagram is the dual graph of the Delaunay triangulation. The circumcenter of each triangle of a Delaunay triangulation constitutes the vertex of the dual Voronoi diagram.
- In the second approach, the smoothing domain is built by first constructing a background of 3-node triangles. This can be generated by a lot of CAD tools or mesh generator for FEM [261]. Then the smoothing domain for each node is built by connecting the centroids of the triangles and the middle points at the three edges of the triangle.

Figure 5.2 shows a comparison of the two types of smoothing cell Ω_k obtained for the same node x_k.

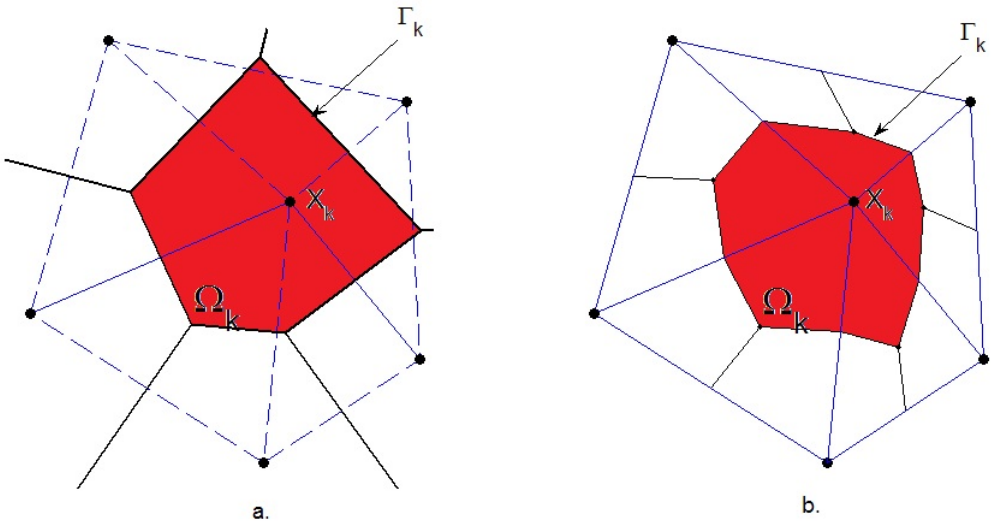


Figure 5.2: Smooth integration nodal domain(red) built by: a. Voronoi diagram; b. Triangle centroid method

5.2 Meshfree based on local weak form

The meshless local Petrov-Galerkin (MLPG) developed by Alturi [2–7] uses a local weak formulation of the governing PDE, defined on overlapping local support domains (figure 5.3). It uses the Petrov-Galerkin formulation with the weight function defined over support domain can be chosen differently from the SFs. The integration is carried out over local quadrature domain which has to be predefined for each node. Simple shape as rectangular and circle domains are usually used for 2D problems, and sphere or parallepiped shapes are used in the case of 3D problems. The local domain and quadrature domain can coincide but this not a criteria for accuracy. The general flowchart for the MLPG procedure is given in figure 5.4.

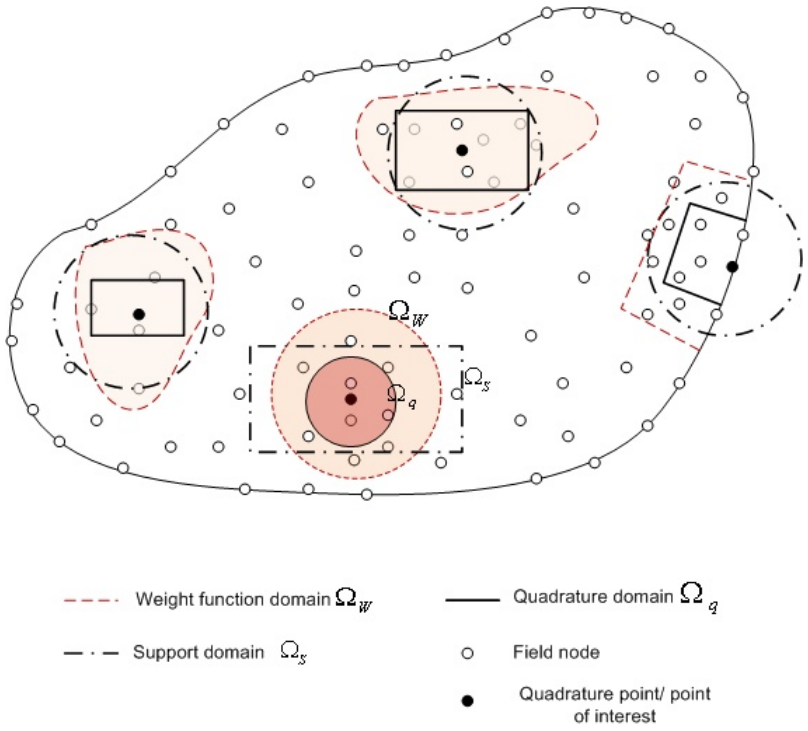


Figure 5.3: Problem domain represented by the MLPG

5.2.1 MLPG Formulation

We consider the strong formulation of the previous elastic beam problem in tensor notation

$$\sigma_{ij,j} + b_i = 0 \tag{5.47}$$

$$u_i = \bar{u}_i \quad on \ \Gamma_u \tag{5.48}$$

$$\sigma_{ij} n_j = \bar{t}_i \quad on \ \Gamma_t \tag{5.49}$$

Taking the local weak formulation using the weight residual method gives

$$\int_{\Omega_q} W_I(\sigma_{ij,j} + b_i) d\Omega = 0 \tag{5.50}$$

W_I is the weight function expressed or centered at the node x_I. Applying the formulation for each node gives rise to the system of equations. When the local residual formulation is used, the compatibility of the SFs over the whole domain is not compulsory. The requirement is reduced to the field function. As long as it is $C^1 \ \forall x_I \in \Omega_q$, the SF $\in H^1$. In a sense only "local compatibility" is required to ensure the existence of the solution.
Using integration by parts on equation 5.50 gives

$$\int_{\Omega_q} W_I \sigma_{ij,j} d\Omega = \int_{\Gamma_q} W_I n_j \sigma_{ij} d\Gamma - \int_{\Omega_q} W_{I,j} \sigma_{ij} d\Omega \tag{5.51}$$

Where n_j being the j^{th} component of the vector of the unit outward normal to the boundary Γ. Substituting equation 5.51 into equation 5.50 gives the following local weak-formulation

$$\int_{\Gamma_q} W_I \sigma_{ij} n_j d\Gamma - \int_{\Omega_q} W_{I,j} \sigma_{ij} d\Omega + \int_{\Omega_q} W_I b_i d\Omega = 0 \tag{5.52}$$

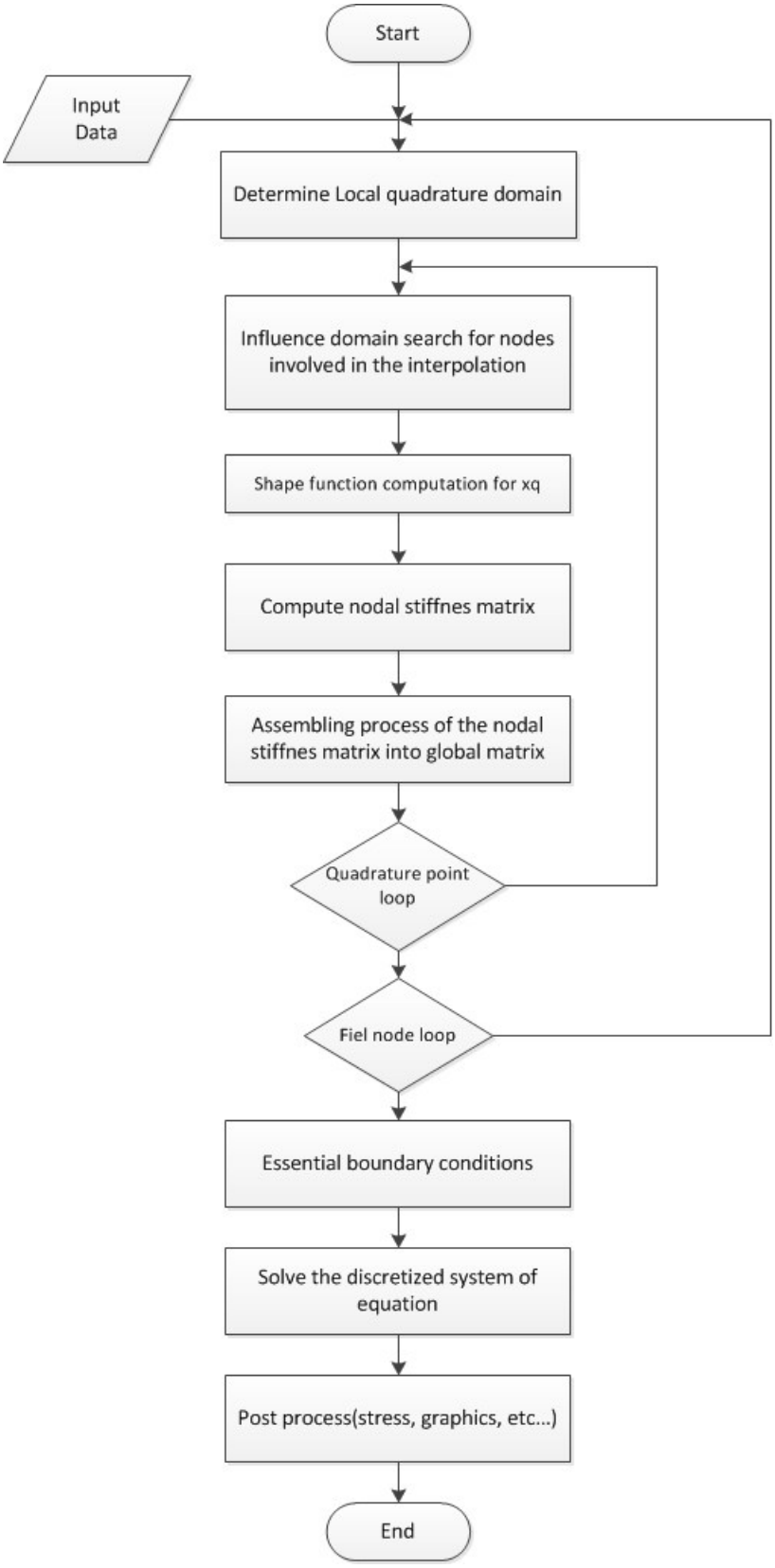


Figure 5.4: General flowchart of the MLPG method

The local quadrature domain $\Gamma_q = \Gamma_{qu} \cup \Gamma_{qt} \cup \Gamma_{qi}$. Γ_{qi} is the internal boundary of the quadrature domain that does not intersect with the boundary domain Γ. Γ_{qu} and Γ_{qt} represents the essential and natural boundary that intersects with the quadrature domain respectively. Thus, the equation 5.52 can be developed

$$\int_{\Gamma_{qi}} W_I \sigma_{ij} n_j d\Gamma + \int_{\Gamma_{qu}} W_I \sigma_{ij} n_j d\Gamma + \int_{\Gamma_{qt}} W_I \sigma_{ij} n_j d\Gamma - \int_{\Omega_q} W_{I,j} \sigma_{ij} d\Omega + \int_{\Omega_q} W_I b_i d\Omega = 0 \tag{5.53}$$

In the case where the local quadrature domain is fully within the interior domain Ω, we have

$$\int_{\Gamma_{qi}} W_I \sigma_{ij} n_j d\Gamma - \int_{\Omega_q} W_{I,j} \sigma_{ij} d\Omega + \int_{\Omega_q} W_I b_i d\Omega = 0 \tag{5.54}$$

Because the Petrov-Galerkin method is used, the weight and trial functions can be chosen to be different. Selecting for W_I to vanish on Γ_{qi} helps to simplify the formulation of equation 5.53.

$$\int_{\Gamma_{qu}} W_I \sigma_{ij} n_j d\Gamma + \int_{\Gamma_{qt}} W_I \sigma_{ij} n_j d\Gamma - \int_{\Omega_q} W_{I,j} \sigma_{ij} d\Omega + \int_{\Omega_q} W_I b_i d\Omega = 0 \tag{5.55}$$

and for equation 5.54

$$- \int_{\Omega_q} W_{I,j} \sigma_{ij} d\Omega + \int_{\Omega_q} W_I b_i d\Omega = 0 \tag{5.56}$$

Using the stress-traction on the boundary relation and the traction boundary condition (equation 5.49) on Γ_{qt} gives for equation 5.53

$$\int_{\Gamma_{qi}} W_I t_i d\Gamma + \int_{\Gamma_{qu}} W_I t_i d\Gamma \int_{\Gamma_{qt}} W_I \bar{t_i} d\Gamma - \int_{\Omega_q} W_{I,j} \sigma_{ij} d\Omega + \int_{\Omega_q} W_I b_i d\Omega = 0 \tag{5.57}$$

The MLPG satisfies the problem equation at the node x_I in an integral sense over local quadrature domain. The size of the local quadrature domain determines to some extent, the smoothing of the numerical approximation.

In order to build the discretised system of equations, the domain Ω is represented by a constellation of nodes. Using MF SFs for the displacement field, the approximate displacement $u^h(x)$ is given by equation 5.10. The local matrix of the weight functions W_I is given by

$$W_I = W(x, x_I) = \begin{bmatrix} W(x, x_I) & 0 \\ 0 & W(x, x_I) \end{bmatrix} \tag{5.58}$$

and $W_{I,j}$ the matrix collecting the derivatives of the weight functions

$$W_{I,j} = W_{,j}(x, x_I) = \begin{bmatrix} W_{,x}(x, x_I) & 0 \\ 0 & W_{,y}(x, x_I) \\ W_{,y}(x, x_I) & W_{,x}(x, x_I) \end{bmatrix} \tag{5.59}$$

The traction vector is

$$t = \begin{Bmatrix} t_x \\ t_y \end{Bmatrix} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \end{Bmatrix} = n_{(2\times3)} D_{(3\times3)} B_{(3\times2n)} u_{(2n\times1)} \tag{5.60}$$

n is the vector of the unit outward normal on the boundary. Using the strain-displacement relation and the constitutive equation and replacing this in equation 5.57 gives

$$\int_{\Omega_q} W_{I,j}^T DBud\Omega - \int_{\Gamma_{qi}} W_I^T nDBud\Gamma - \int_{\Gamma_{qu}} W_I^T nDBud\Gamma = \int_{\Gamma_{qt}} W_I^T \bar{t} d\Gamma + \int_{\Omega_q} W_I^T bd\Omega \tag{5.61}$$

This leads to the typical matrix form of equation

$$(K_I)_{2\times 2n}(u)_{2n\times 1} = (f_I)_{2\times 1} \tag{5.62}$$

K_I is called the nodal stiffness matrix for the I^{th} given by

$$K_I = \int_{\Omega_q} W_{I,j}^T DBd\Omega - \int_{\Gamma_{qi}} W_I^T nDBd\Gamma - \int_{\Gamma_{qu}} W_I^T nDBd\Gamma \tag{5.63}$$

f_I is the nodal force vector collecting the contributions from body forces and traction on the boundary (natural boundary condition).

$$f_I = \int_{\Gamma_{qt}} W_I^T \bar{t}\mathrm{d}\Gamma + \int_{\Omega_q} W_I^T b\mathrm{d}\Omega \tag{5.64}$$

In the case of a fully interior quadrature domain, equation 5.54, we have the simpler form

$$K_I = \int_{\Omega_q} W_{I,j}^T DB\mathrm{d}\Omega - \int_{\Gamma_{qu}} W_I^T nDB\mathrm{d}\Gamma \tag{5.65}$$

$$f_I = \int_{\Omega_q} W_I^T b\mathrm{d}\Omega \tag{5.66}$$

At this stage, Gauss integration can be used to obtain the discretised system of equations for the integrals

$$\begin{aligned} K_I = &\sum_{k=1}^{n_g} w_k W_{I,j}^T(x_{gk})\mathrm{DB}(\mathrm{x_{gk}}) \left|J_q^D\right| - \\ &\sum_{k=1}^{n_g t} w_k W_I^T(x_{gk})\mathrm{nDB}(\mathrm{x_{gk}}) \left|J_{qi}^B\right| - \sum_{k=1}^{n_g t} w_k W_I^T(x_{gk})\mathrm{nDB}(\mathrm{x_{gk}}) \left|J_{qu}^B\right| \end{aligned} \tag{5.67}$$

The force vector becomes

$$\mathrm{f_I} = \sum_{k=1}^{n_g t} w_k W_I^T(x_{gk})\bar{t} \left|J_{qt}^B\right| + \sum_{k=1}^{n_g} w_k W_I^T(x_{gk})\mathrm{b} \left|J_q^D\right| \tag{5.68}$$

n_g is the total number of Gauss points within the quadrature domain and n_{gt} the number of Gauss points used for the curve integral of boundary domain. w_k is the Gauss weight for the Gauss point x_{gk}. We apply the equation 5.62 for each node of the domain Ω leading to 2N independent linear equations. Then, using the global numbering system, one can finally assemble these equations to

obtain the global system of equations.

$$\begin{bmatrix} K_{11} & K_{11} & \cdots & K_{(1)(2N-1)} & K_{(1)(2N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{(2I-1)(1)} & K_{(2I-1)(2)} & \cdots & K_{(2I-1)(2N-1)} & K_{(2I-1)(2N)} \\ K_{(2I)(1)} & K_{(2I)(2)} & \cdots & K_{(2I)(2N-1)} & K_{(2I)(2N)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{(2N)(1)} & K_{(2N)(2)} & \cdots & K_{(2N)(2N-1)} & K_{(2N)(2N)} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \vdots \\ u_I \\ v_I \\ \vdots \\ u_N \\ v_N \end{Bmatrix} = \begin{Bmatrix} f_{1x} \\ f_{1y} \\ \vdots \\ f_{Ix} \\ f_{Iy} \\ \vdots \\ f_{Nx} \\ f_{Ny} \end{Bmatrix} \tag{5.69}$$

For a 2D problem, this leads to a $2N \times 2N$ matrix system and a $2N \times 1$ right hand vector. The assembling process of MLPG, similar to the process used in FDM, is different from the FEM and the global weak form based meshfree methods. Here the nodal matrix is stacked together row-by-row (for each point of interest) in order to build the global matrix. To perform the integration over local quadrature domain Ω_q for a considered point of interest x_I, we need to build a quadrature domain Ω_q. Then for each quadrature point x_q within this quadrature domain, the meshfree shape functions are built to obtain the integrand in equation 5.61. Thus Meshfree local weak form is slightly more complicated in the management of local domains. For a field node x_i, we have the following different domains as shown in figure 5.3:

- The local quadrature domain Ω_q of size r_q.
- The local domain for support of the weight function Ω_w of size r_w.
- The local support domain Ω_s for x_q of size r_s.

An important condition is that r_q<r_w. However, as already mentioned in the case where the chosen weight function vanishes on the boundary γ_{qi}, this leads to the simplified local weak form 5.55. One can chose $r_q = r_w$ to simplify the local weak form. For a point of interest x_I, a simple way to define the size of the local quadrature domain r_q or the support domain r_s is

$$r_q = \alpha_q d_{cI} \quad \text{and} \quad r_s = \alpha_s d_{cI} \tag{5.70}$$

d_{cI} is taken as the nodal spacing average near the point of interest x_I. The coefficient α_s and α_q are dimensionless parameter which allows us to control the local subdomain size. Rectangular domains are simpler and easier to use. The dimension of quadrature domain is determined

$$r_{qx} = \alpha_{qx} d_{cx} \quad \text{and} \quad r_{qy} = \alpha_{qy} d_{cy} \tag{5.71}$$

α_{qx} and α_{qy} are dimensionless sizes of the quadrature domain respectively in the x- and y- directions. d_{cx} and d_{cy} are the nodal spacing in each direction.

5.2.2 System stiffness matrix properties

Because MLPG uses compact support domain for W, the System matrix is sparse. With a proper system of numbering for the field nodes, it can produce a banded K matrix (equation 5.63). The matrix K is also asymmetric. The Petrov-Galerkin formulation is based on the choice of different

test and weight functions. On top of this, different local domains can be of different sizes. Local quadrature domain Ω_q and Ω_s can be different as well from one node to another. Thus for 2 different nodes of interest the local domain in equation 5.63 will be different

$$\int_{\Omega_q^{(k)}} \mathrm{W}_{\mathrm{k,j}}^{\mathrm{T}} \mathrm{DB_l d}\Omega \neq \int_{\Omega_q^{(l)}} \mathrm{W}_{\mathrm{l,j}}^{\mathrm{T}} \mathrm{DB_k d}\Omega \tag{5.72}$$

$\Omega_q^{(k)}$ and $\omega_q^{(l)}$ are local quadrature domains for k^{th} and l^{th} nodes respectively. $W_{k,j}$ and $W_{l,j}$ are the matrices of the derivative of the weight functions for k^{th} and l^{th} nodes. Finally B_l and B_k are the strain matrix at each node. In equation 5.63 the boundary integration members are also asymmetric. For the same reason as before, the local subdomains for different nodes can vary, leading to an unsymmetrical boundary matrix. Thus in MLPG the stiffness system matrix is generally asymmetric, banded and sparse which increases the computational cost compared to the EFG method.

5.2.3 Weight function

In meshfree based on local weak formulation with the Petrov-Galerkin method, the weight function is chosen differently from the trial function. In fact, such meshfree method is no more than a local weight residual method and the weight function does play an important part in the performance of the method. Functions decreasing in magnitude when the distance from the center of the support domain increases give better results [5, 7]. In theory any function may be used, but for practical considerations, functions depending only on the distance $\|x_I; x\|$ are preferred. Functions, which vanish on the Γ_{qi}, simplify the formulation. Atluri [4] originally used spline function in the original MLPG formulation such the as the W1 (cubic) or the 4th order spline (W2). They have been proven to be the best trade-off in terms of simplicity and accuracy. For the quartic spline, the function vanishes if $r_q = r_w$ and for W1 $r = r_q$.

5.2.4 Boundary conditions

Local weak form based meshfree methods such as the MLPG are assembled node by node. For each node, there are n_f rows in the global stiffness matrix as shown in (equation 5.69 and the global force vector. Due to the nature of the MLPG system equation similar to the FDM, the direct interpolation method [7, 196] can be used to enforce the essential boundary conditions. For the I^{th} node on the boundary with prescribed displacement(see equation5.48), we have the following two linear equations

$$\begin{cases} u_I^h = \phi_1 u_1 + \phi_2 u_2 + \cdots + \phi_n u_n = \bar{u}_I \\ v_I^h = \phi_1 v_1 + \phi_2 v_2 + \cdots + \phi_n v_n = \bar{v}_I \end{cases} \tag{5.73}$$

Following the MLPG assembly process described in section 5.2.1, and replacing the equations for all the essential boundary node x_I, we obtain the final modified global system of equations $K_{2N\times1}U_{2N\times1} = F_{2N\times1}$.

5.2.5 Efficiency of local weak forms based meshfree methods

The main advantage of local weak form based meshfree methods is the absence of a background mesh for the integration. This confers a true meshless nature to the method. For the same num-

ber of DOF, MLPG is computationally more expensive than FEM and global weak form based methods. The increase in the computational cost is mainly due to the following reasons

- The process in building meshfree shape function is more costly than the one for FEM for a relative small benefit in term of overall accuracy.
- The numerical integration in MLPG is more complicated due to:
  - Local quadrature domains intersecting with global boundary domain with natural or essential boundary conditions need to be mapped to standard Gauss domain in order applied numerical quadrature. The intersected portions of surfaces or curves need to be determined in order to compute the boundary integral in equation 5.61.
  - The overlapping of quadrature domains needs to be optimised in order to reduces the computational charge.
- Solving the asymmetric global stiffness system matrix is computationally more expensive that the one produced in EFG.

5.3 Support and influence domain

The support domain or influence domain plays an important role in meshfree methods. The support domain for a field point of interest $x_q = (x, y, z)$ is defined as the domain centered around this point where all the nodesx_i information located within this domain are used to compute the information at x_q.
On the other hand, the influence domain is a domain centered at a node which defines the area (points within the domain) upon which it has an influence.

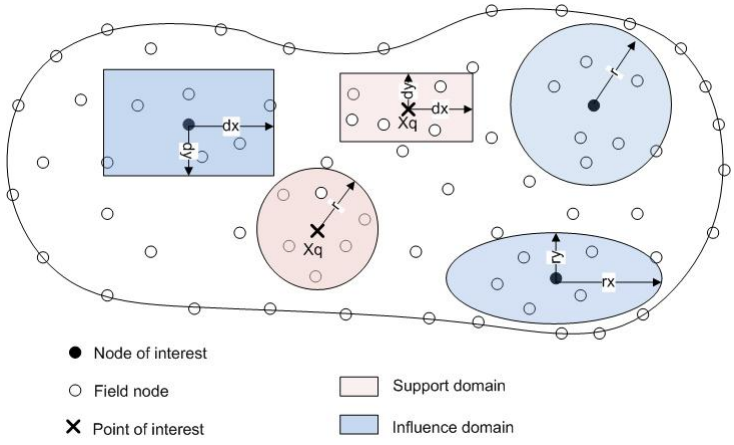


Figure 5.5: Examples of different type of support and influence domain

The influence domain is associated with a node whereas a support domain is built at a point of interest x_q. The latter determines which nodes or points are used to build the approximation at x_q. In EFG, support domains are built around Gauss quadrature points.
To illustrate how differently these two domains work and to show the limitations of the support domain, we consider the following simplified case. Let us consider a domain $\Omega \subset \mathbb{R}^3$. We consider the set of nodes (particles) $X_i \in \Omega$ with $i \in 1, ..., 7$ with $X_i = (x_i, y_i, 0) \in \Omega$ (see figure 5.6). Let u be an unknown continuous function: $u : \Omega \to \mathbb{R}$. In the process of building an

approximation u^h at node 1, we first consider the support domain there. The size of the support domain is set to fulfil the compact requirements of the meshfree formulation. Nodes 2,3,4,5 and 6

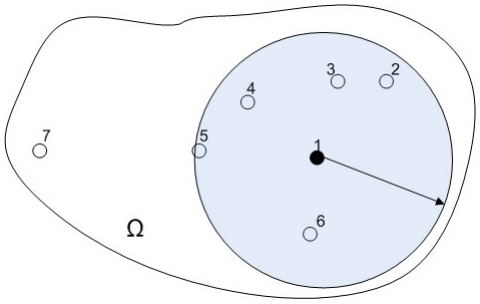


Figure 5.6: Support domain built around node 1

are considered to build an approximation of u at the node 1 with $u_1^h = \sum_{i=2}^{6} \phi_i u_i$.
Now let us approximate the field function u at node 1 but with the use of influence domains. The influence domains are drawn on figure 5.5. Nodes which are far from each other (7 and 1) can still have an influence on each other whereas nodes that are very close (node 1, 2, 5 and 6) can be ignored when building the approximation. This time, only nodes 3, 4 and 7 are used to build the

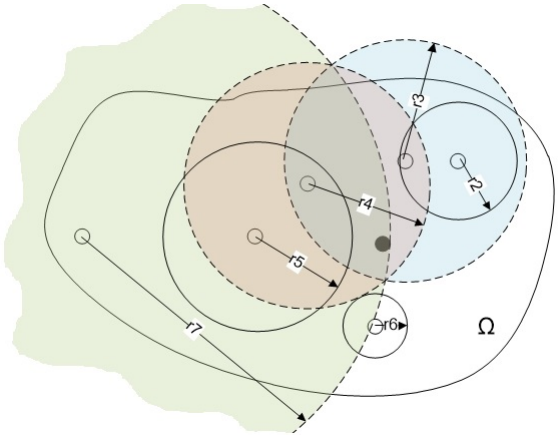


Figure 5.7: Influence domain of the node covering node 1

approximation. Node 7 which is the furthest from 1 still has an influence on the approximation at node 1. This is of great interest in case of geometric, material or field function discontinuities or when the nodes are isolated like node 7.
We now consider another set of seven nodes but with a crack within the domain Ω. Consider an approximation based on a support domain at the node 1 (figure 5.8). There is a discontinuity within the support domain of node 1, which makes the computation of the shape function difficult. This requires special treatment and is generally avoided (see section 5.6). It is for this reason that the use of influence domain described in figure 5.9 can provide a better solution. The influence domain solves many problems introduced by the use of the support domain. In general, the influence domain is mostly equivalent to the support domain for isolated nodes. Various shapes of the influence domain, such as discs, rectangles for 2D application and sphere or parallepideds for 3D can be used. Influence regions can also be defined with tree structures linked to domain partitions for large scale problems.
As shown in section 4.3, the accuracy of the interpolation of meshfree shape functions is influenced by the number of nodes in the support domain. For optimum results we can define the support size

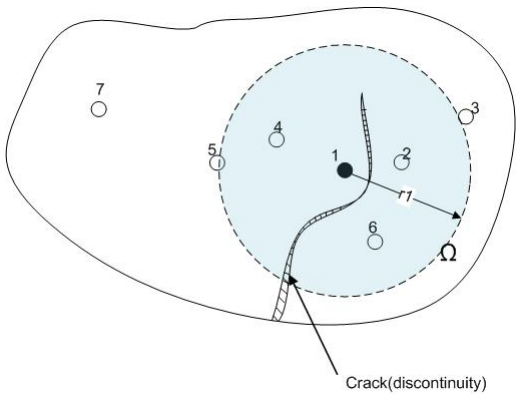


Figure 5.8: Support domain at node 1 with geometric discontinuity

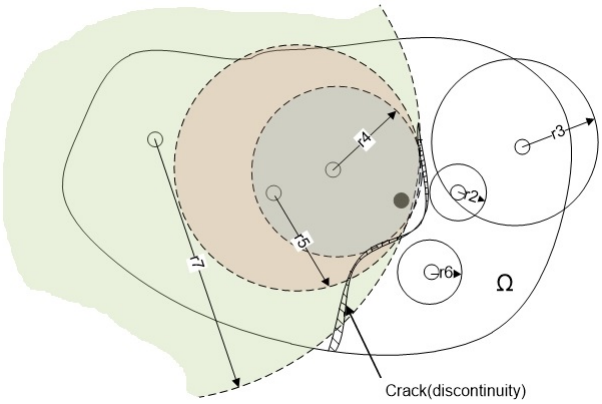


Figure 5.9: Influence domain of the node covering node 1 and a crack within the domain Ω

or coverage as

$$d_s = \alpha_s d_c \tag{5.74}$$

Where α_s is a dimensionless parameter controlling the support size of the domain and d_c being the characteristic length. d_c is directly related to the nodal spacing in the surrounding of the sampling point. α_s can be predetermined in the case of uniform nodal distribution or can vary to optimise the selection of nodes within the support domain. For a 2D problem, the average nodal spacing can be expressed as such [159]

$$d_c = \frac{\sqrt{A_s}}{\sqrt{n_{A_s}} - 1} \tag{5.75}$$

with A_s the area covered by a support domain of dimension d_s. n_{A_s} is the number of nodes. The procedure to determine the support domain size can be broken down into the following steps

1. Estimate d_s at the sampling point x_q
2. Compute A_s
3. Compute d_c
4. Using equation 5.74, obtain d_s for a given α_s.

In 3D cases A_s can be substituted by the volume V_s covered by the support domain.

5.4 Neighbour nodes search

In meshfree methods, the approximation is "local" and only used on a finite set of neighbour particles, often called the nearest neighbouring particle (NNP). In meshfree methods and especially in particle methods, the position of the nodes is not always known in advance and can vary with time. Thus the NNPs are constantly changing which requires an algorithm to determine each of these points at each approximation step. Such algorithms are called Nearest Neighbouring particle Searching (NNPS).
The simplest method is to look through the whole set of nodes used to discretise the domain and check if they are included in the support domain of the node of interest. It is sometimes called all paired search algorithm (APS) for SPH and consists in running a double cycle through the nodes to examine all possible pairs. This method is time consuming with a number of operations of order $N_{op}O(n^2)$ (n being the total number of nodes). Another method is the Linked-list algorithm (LLS).
This technique [117] requires the use of a Cartesian grid superimposed over the discretised domain by particles (see 5.10). Each cell of the grid is then used to store information on the positions of the particles. A cycle is used to find the corresponding cell for each particle. Subsequently, the search for neighbour particles is constrained to the particles within the cell linked to the particle of interest or within the cells adjacent to it. This algorithm is much faster than the ALS but more memory is required to store the grid and the data structure of nodes for each cell. The number of operations is $N_{op} = O(n)$ or more exactly $N_{op} = O(n\log(n))$.

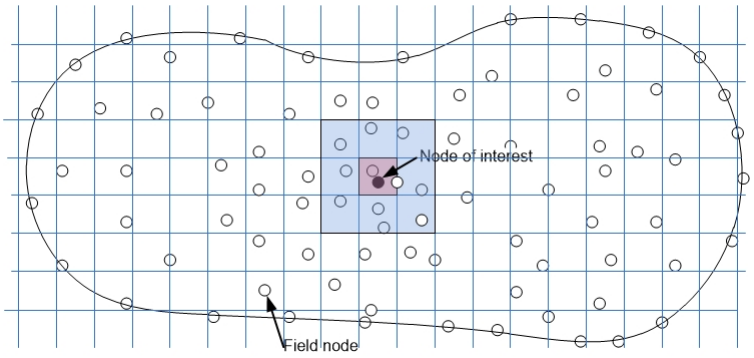


Figure 5.10: Linked-list particle searching algorithm grids. The particle in the highlighted cells are considered for interactions

Other techniques such as the Bucket technique [159], trees, k-d trees in 2D for instance and k-nearest neighbour(k-NN) search are also effective ways to determine nodes within each influence domain. Such techniques that rely on trees with spatial decomposition are especially efficient for real geometry problems where node searching can be done concurrently while building SFs, considerably reducing the computational time. Moreover the re-organisation of nodes in one particular region only covered by a set of bucket or leaf can be done without impacting the rest of the data structure. This leads to tremendous benefits for geometric moving boundary problems.

5.5 Benchmark problem: 2D Elasto-static cantilever Beam

A recurrent and often encountered structure in MEMS [140] is the beam. A beam is a solid whose dimension in one direction is much larger than the two others. The cantilever microswitch

is a widely used structure in MEMS. It can be simplified into a cantilever model. There is a colossal amount of research devoted to the study of such structures and FEM is the method used in engineering problems involving devices based on this structure. In MEMS, beams are involved in many mechanical parts found in sensors and actuators. It qualifies as a good benchmark problem to investigate the use of meshfree methods for elasto-static problem in MEMS design. A beam can vary from a simple 1D problem to a very complex 3D problem with laminating material, non homogeneous and thermal dependent material properties. The implementation of the following problems was carried out in Matlab code and C++.
We consider the benchmark problem which features a cantilever beam subjected to a parabolic traction at the free right end as shown on figure 5.11. We assume the beam to have a unit thickness.

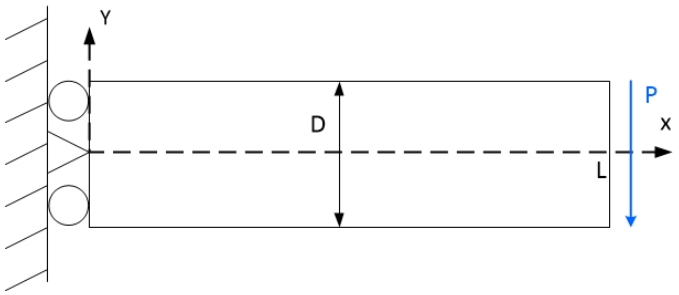


Figure 5.11: 2D cantilever beam with parabolic traction at his right free end

The problem is a typical plane stress problem where the solution stress is $\sigma = \{\sigma_{xx}\sigma_{yy}\sigma_{xy}\}$. The formulation of the problem is given in section 5.1 page 93. The analytical displacement solution for the linear elastic cantilever beam under constant load is given by

$$\begin{aligned} u(x,y) &= \frac{-Py}{6EI}\left((6L-3x)x + (2+\nu)\left(y^2 - \frac{D^2}{4}\right)\right) \\ v(x,y) &= \frac{P}{6EI}\left(3\nu y^2(L-x) + (4+5\nu)\left(\frac{D^2 x}{4}\right) + (3L-x)x^2\right) \end{aligned} \tag{5.76}$$

$I = \frac{D^3}{12}$ is the moment of inertia for a beam with a rectangular cross-section and unit thickness. The normal stress on the cross-section of the beam is given

$$\sigma_{xx}(x,y) = \frac{P(L-x)y}{I} \tag{5.77}$$

Then normal stress $\sigma_{yy} = 0$ in y-direction null and the shear stress on the cross-section is given by

$$\sigma_{xy} = \tau_{xy}(x,y) = \frac{P}{2I}\left[\frac{D^2}{4} - y^2\right] \tag{5.78}$$

The loading is taken to be P=1000 N. The Poisson ratio $\nu = 0.3$. The height of the beam D=15, the length L=60. The essential boundary condition prescribed on the left boundary side of the beam(x=0) is given using the equations 5.76. At the right end of the beam(x=L), we apply the traction force $t_{xy}|_{x=L}$ using the analytical equation 5.78 and a parabola distribution on the cross section. The shear stress on the cross section of the beam is given by

$$t_{xy}(x,y) = \frac{P}{2I}\left[\frac{D^2}{4} - y^2\right] \tag{5.79}$$

Figure 5.12 shows the deflection of beam simulated using EFG with a regular distribution of nodes

21×9. The real displacement is amplified by a factor 1000 to visualize the deflection. The EFG solution (red) is compared with the analytical solution (blue dash), showing good agreement with the values of the displacement.

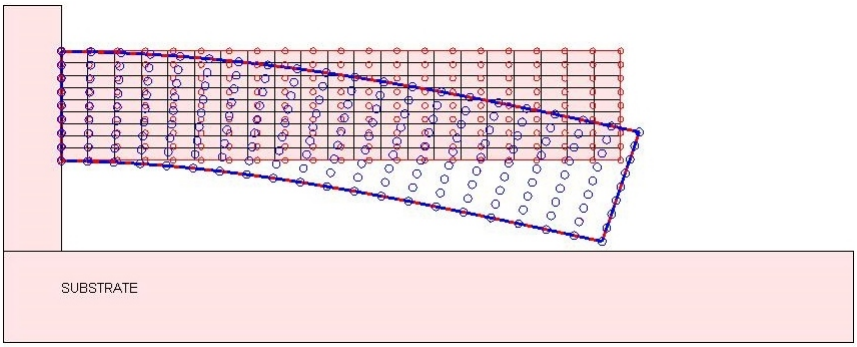


Figure 5.12: Deflection of 2D cantilever beam with parabolic load at the right free end.$L = 60\mu m$, $h = 15\mu m$ 21×9 with $\alpha_s = 2.0$

We use the following displacement norm

$$e_u = \sqrt{\int_\Omega (\boldsymbol{u^h} - \boldsymbol{u_a})^T \cdot (\boldsymbol{u^h} - \boldsymbol{u_a}) d\Omega} \tag{5.80}$$

and as the strain is usually more critical in simulation, we use the energy norm

$$e_\epsilon = \sqrt{\frac{1}{2}\int_\Omega (\boldsymbol{\epsilon^h} - \boldsymbol{\epsilon_a})^T D (\boldsymbol{\epsilon^h} - \boldsymbol{\epsilon_a}) d\Omega} \tag{5.81}$$

We also use the L_2 relative error norm for the cross section normal stress σ_{xx} and shear stress σ_{xy}. Figure 5.13 shows the convergence for both EFG using penalty and lagrange multipliers to enforce the prescribed nodal displacements. Similar to the FEM, we obtain a better accuracy in

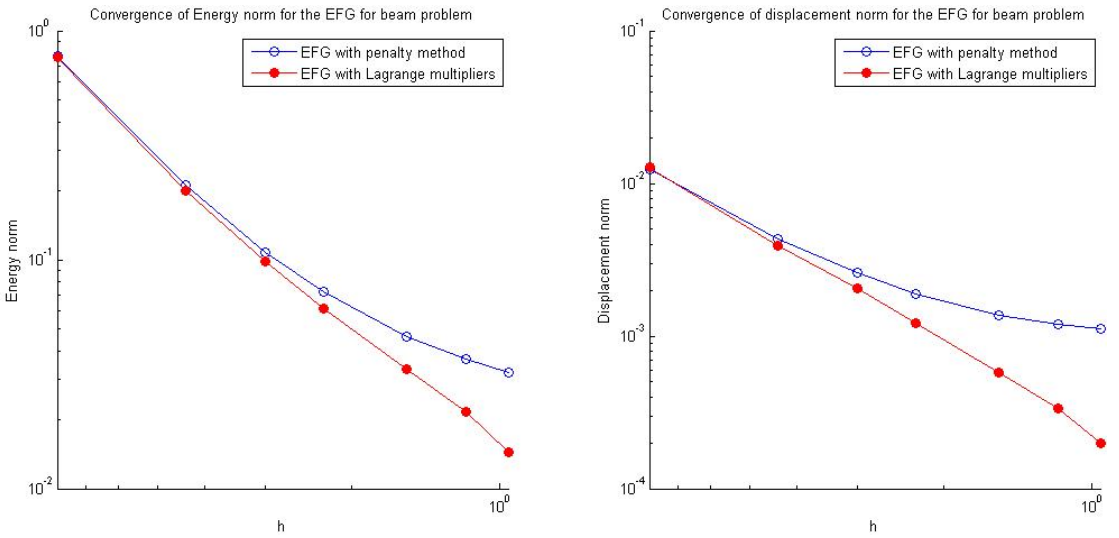


Figure 5.13: Convergence of energy norm and displacement norm for EFG with MLS and cubic spline. 2×2 Gauss points per cell. $\alpha_s = 2.5$

displacement than in strain and stress. This is typically the case for shear stress as shown on figure 5.14.

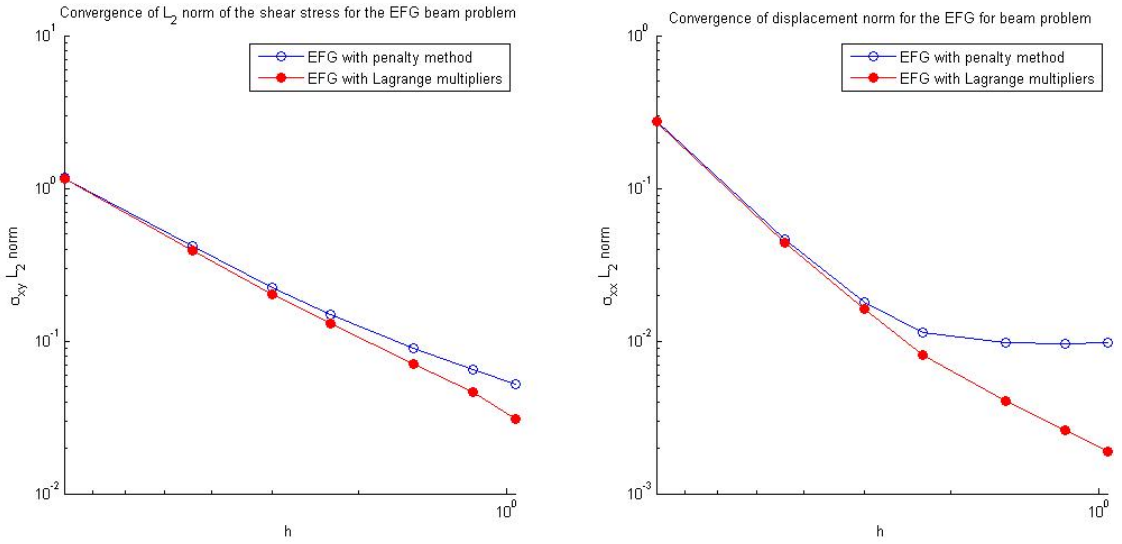


Figure 5.14: Convergence of L_2 error norm for shear σ_{xy} (left) and normal stress σ_{xx} (right) for EFG with MLS and cubic spline. 2×2 Gauss points per cell. $\alpha_s = 2.5$

As the refinement h gets smaller, the penalty method converges slower than the Lagrange multiplier method. This is mainly due to the fact that, as previously stated, the penalty method only enforces the essential boundary conditions in an approximate manner. This discrepancy is more obvious on the displacement and normal stress.

Figure 5.15 shows the convergence of the energy norm and the diplacement norm for RPIM using both using various RBF and CSRBF. Lagrange multipliers are used to enforce the prescribed nodal displacements.

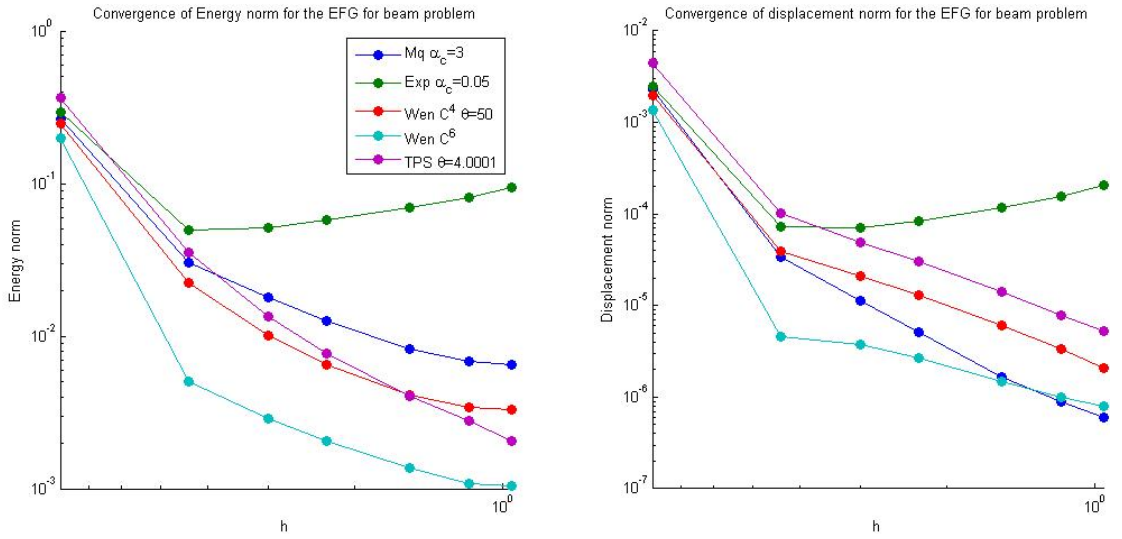


Figure 5.15: Convergence of energy norm and displacement norm for RPIM with different RBF and CSRBF. 2×2 Gauss points per cell. $\alpha_s = 3$

Figure 5.16 shows the convergence of the L^2 norm for the shear σ_{xy} and normal stress σ_{xx}.

The rate of convergence for the energy norm is 1.8 and 1.5 for EFG with the Lagrange multipliers and the EFG with penalty method respectively. The rate of convergence is computed using

$$R = \frac{log\left(\frac{e_{i+1}}{e_i}\right)}{log\left(\frac{d_c^{i+1}}{d_c^i}\right)} \tag{5.82}$$

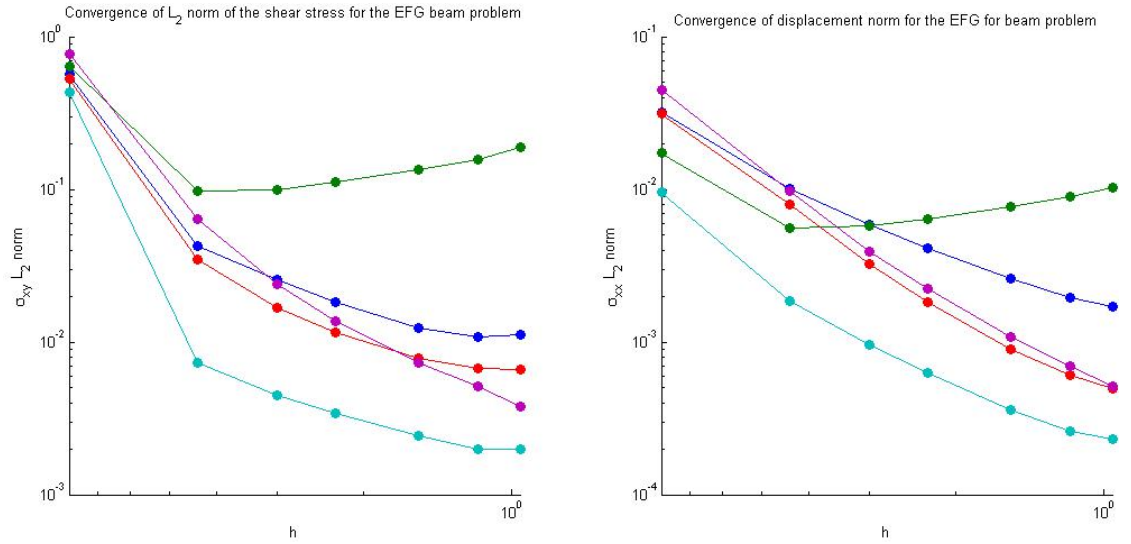


Figure 5.16: Convergence of L_2 error norm for shear σ_{xy} (left) and normal stress σ_{xx} (right) or RPIM with different RBF and CSRBF. 2×2 Gauss points per cell. $\alpha_s = 3$

As expected, the EFG possesses a higher rate of convergence than the FEM with linear piecewise elements. The convergence rate for RPIM is much higher in the first refinement step as shown on figure 5.15. This is interesting given that the rate of increase in computational cost is consistent through the whole refinement as shown in figure 5.17. R_1 is for the first part of h refinement and

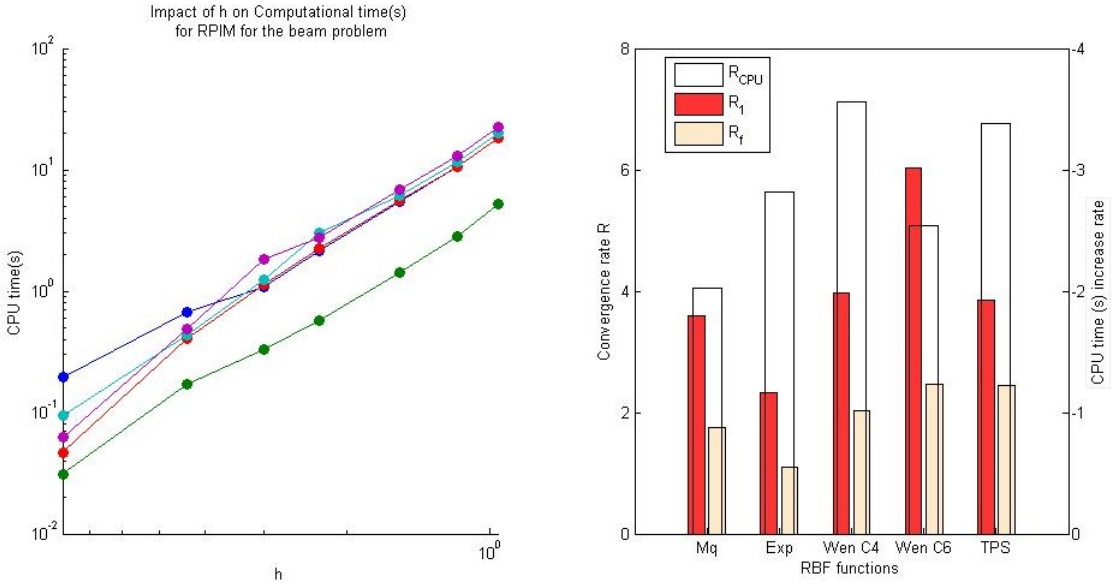


Figure 5.17: Impact of h refinement on the CPU time(s) for RPIM and convergence Rate for different RBF. 21×10 field nodes.2×2 Gauss point per cell.

R_f is the overall convergence rate. R_{CPU} is the rate of increase of computational time(s).
We now study the impact of the support domain size by varying the dimensionless parameter α_s with a regular distribution of nodes 20×10. Figure 5.18 and 5.19 confirms the studies of section 4.3, that a α_s in the range of 2-3 provides optimal results.
Figure 5.20 shows the result using RPIM with different RBF and CSRBF.
Figure 5.21 and 5.23 illustrate the impact of the RBF parameter on the accuracy of RPIM when the support domain is increased.
When the support domain is too small, there are not enough nodes within the support domain to

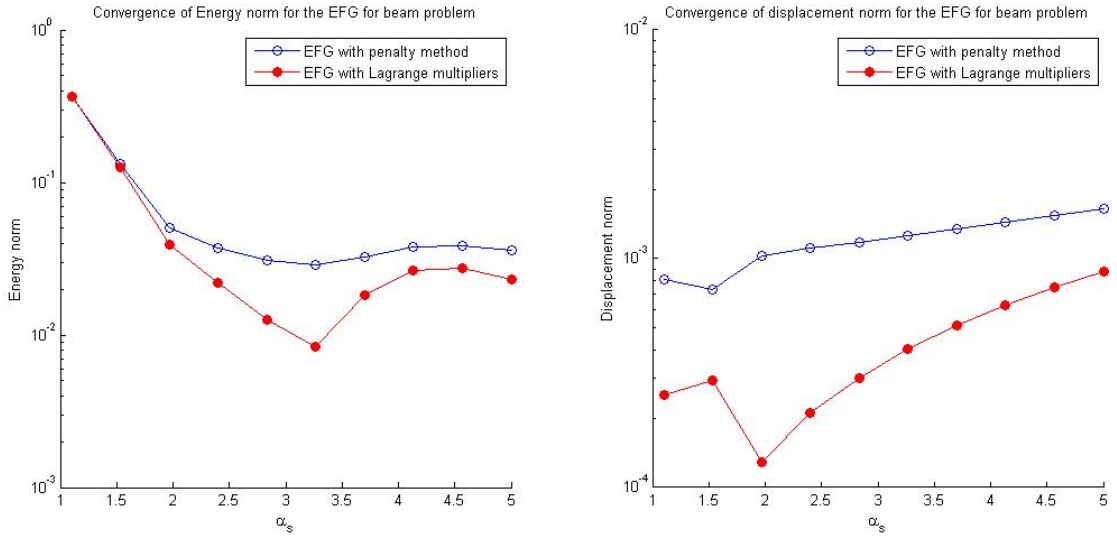


Figure 5.18: Impact of Support domain size on energy norm and displacement norm for EFG with MLS and cubic spline. 2×2 Gauss points per cell

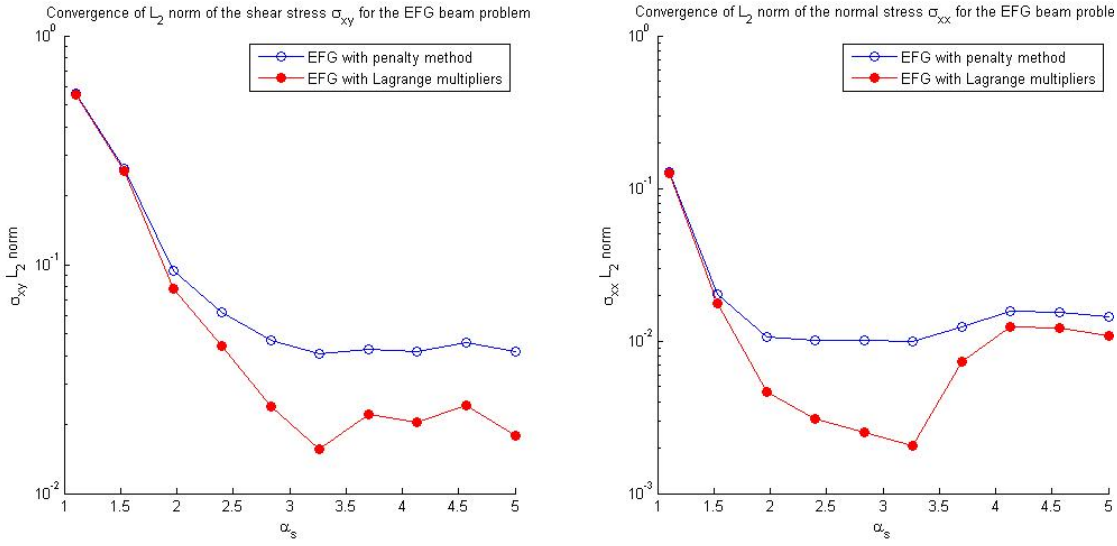


Figure 5.19: Impact of α_s on L_2 error norm for shear σ_{xy} (left) and normal stress σ_{xx} (right) for EFG with MLS and cubic spline. 2×2 Gauss points per cell.

build a proper approximant using either MLS or RPIM. However having too many nodes within a support domain also degrades the quality of the approximation and the condition number of the moment matrix both for MLS and RPIM. Increasing the support domain will also dramatically increase the computational cost of the method.

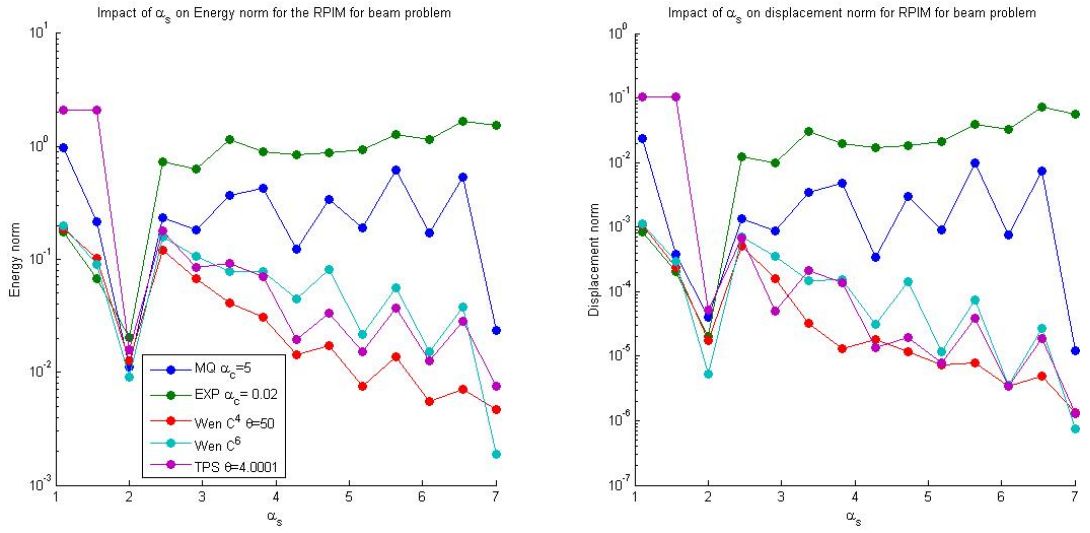


Figure 5.20: Impact of Support domain size on energy norm and displacement norm for RPIM with lagrange multipliers. 4×4 Gauss points per cell

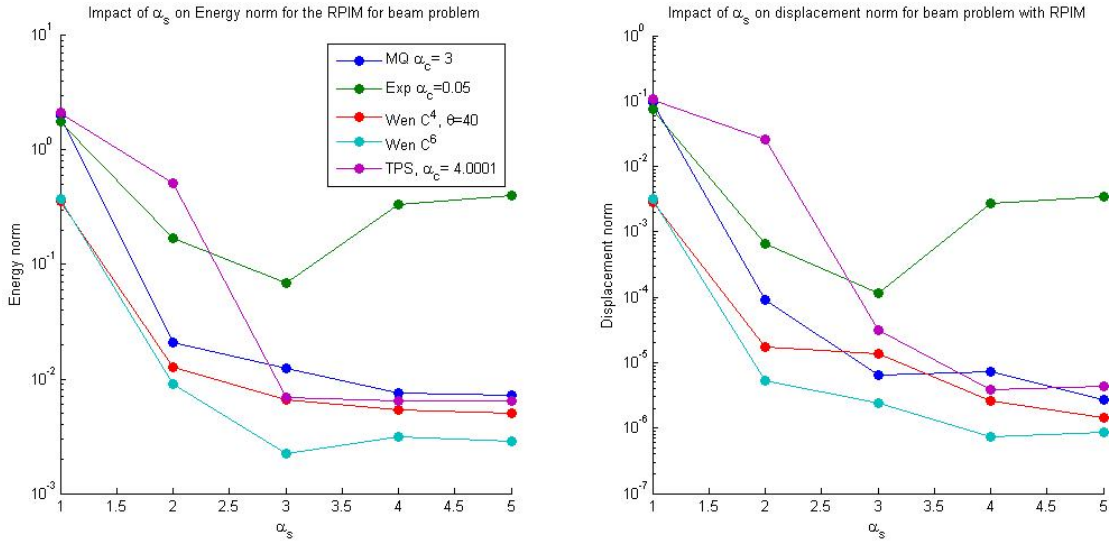


Figure 5.21: Impact of α_s on the energy norm and displacement norm for RPIM. 21×10 field nodes.

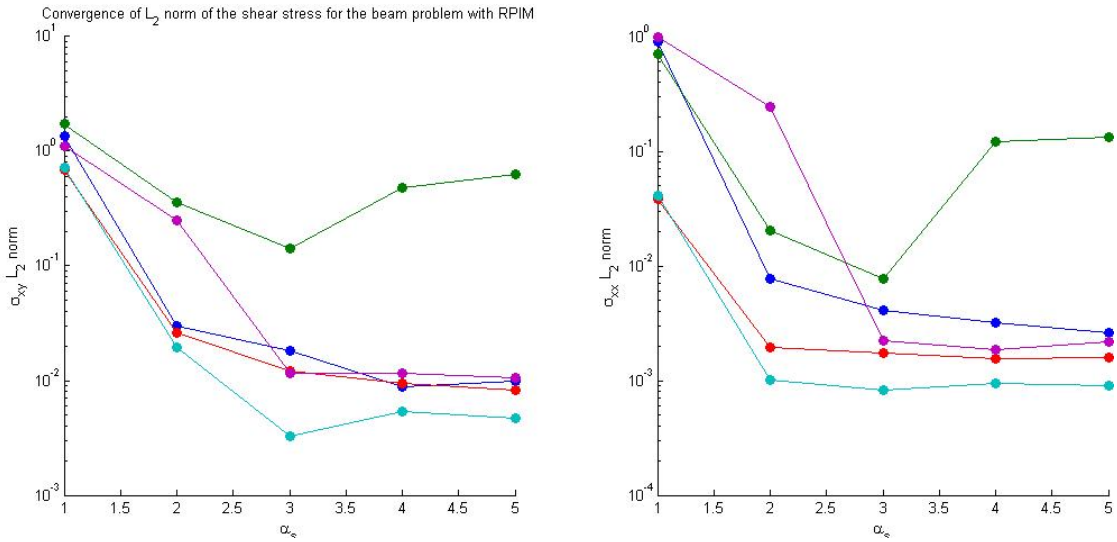


Figure 5.22: Impact of α_s on the L_2 error norm for RPIM. 21×10 field nodes.

Figure 5.22 shows a comparison of the computational time.

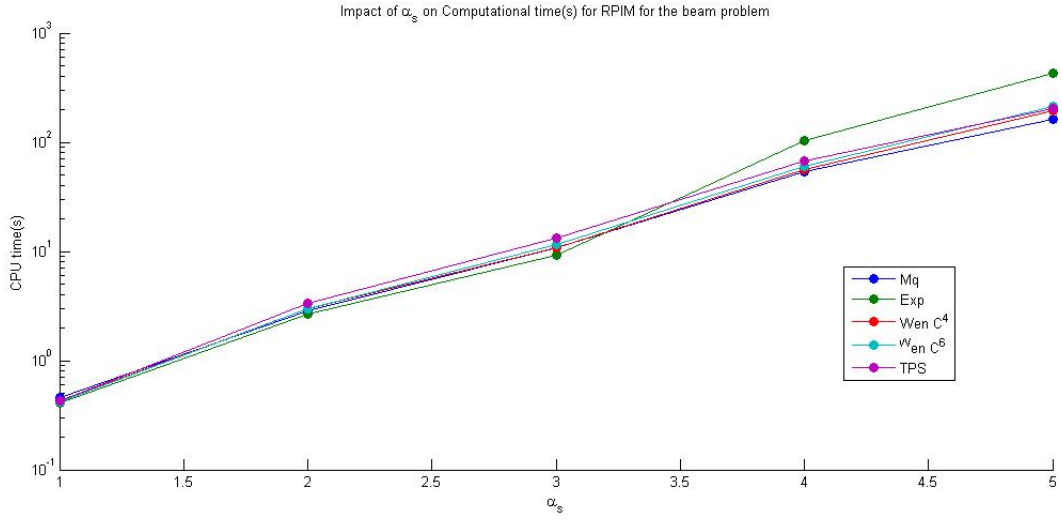


Figure 5.23: Impact of α_s on the CPU time(s) for RPIM. 21×10 field nodes.4×4 Gauss point per cell.

Following the study on optimal parameters for RBF and CSRBF for interpolation accuracy, the beam problem is a good benchmark to characterise the values for elastic problem applications.
The study on parameter α_c confirms the choice of values selected for the parameter in the section on approximation of meshfree SFs. 5.24 5.25 show the result for the RPIM MQ and EXP.

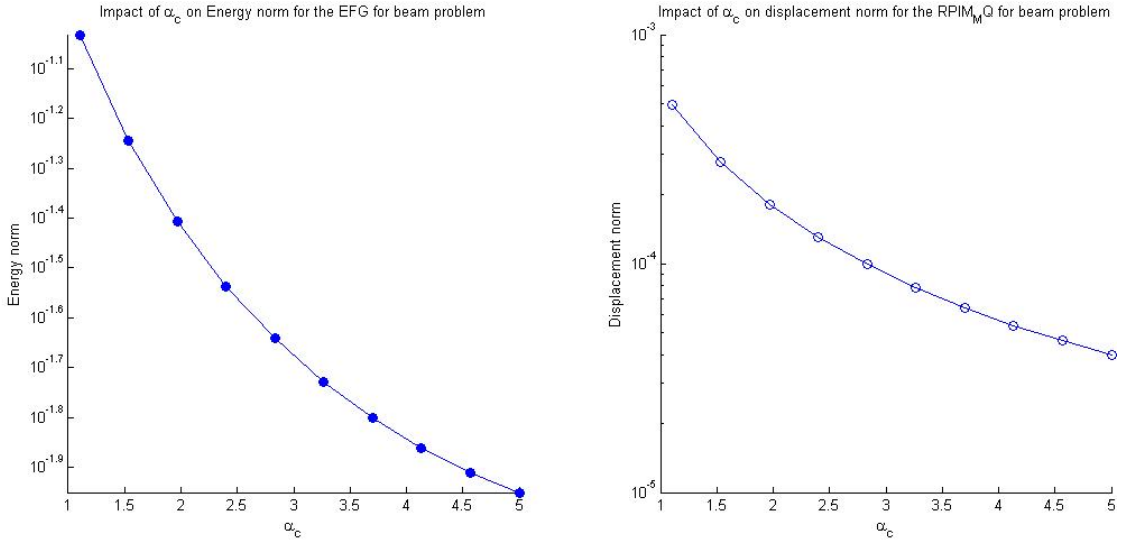


Figure 5.24: Impact of α_c on the energy norm and displacement norm. 21×10 field nodes. $\alpha_c = 3.0$

The optimal choice of parameter α_c for the RPIM TPS is obvious. The optimal α_c seems to narrow down around even values. However the size of the support domain greatly impacts the possible values. For instance figure 5.26 shows the increase in optimal even values for different α_c.
Figure 5.27 shows that the minimum energy error norm is obtained for values such as 2.0001 , 4,0001, 6,0001.
Similar results are also demonstrated in figure 5.28 and 5.29 for CSRBF.
CSRBF are strictly positive definite functions in $\mathbb{R}^d$ with $d <= 3$. They can be constructed to have any desired amount of smoothness $2k$. For $l = \lfloor d/2 \rfloor + k + 1$ with k=0,1,2,3, the Wendland's

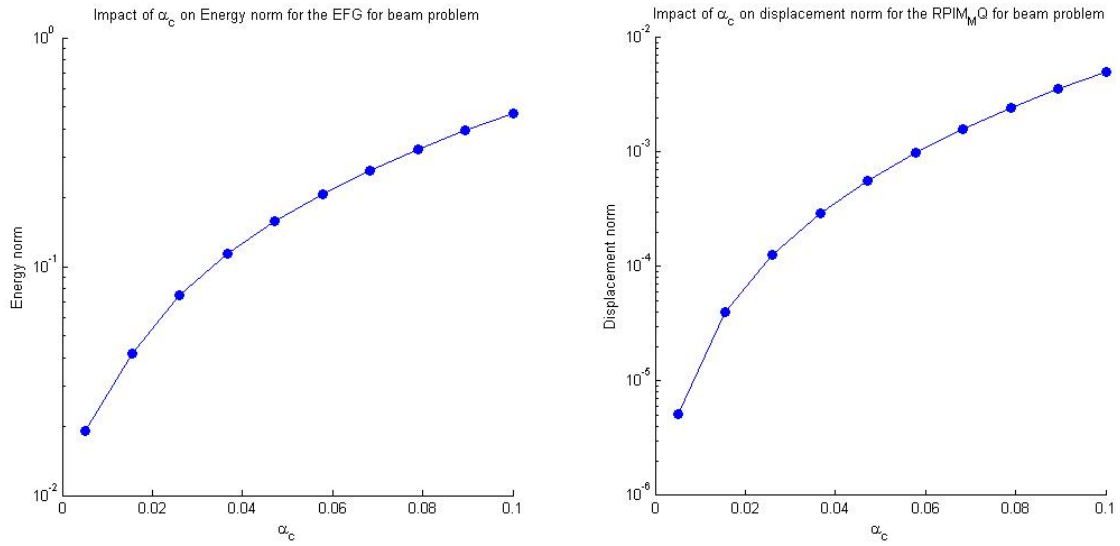


Figure 5.25: Impact of α_c on the energy norm and displacement norm for the RPIM EXP. 21×10 field nodes. $\alpha_c = 3.0$

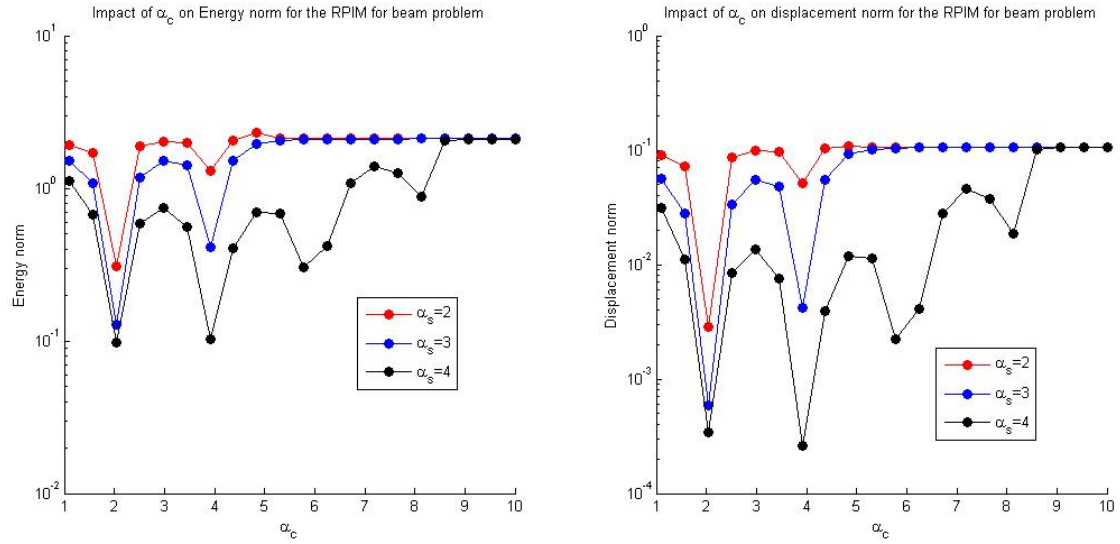


Figure 5.26: Impact of α_c, for several values of α_s, on the energy norm and displacement norm for the RPIM TPS. 21×10 field nodes.

functions $\phi_{d,k}(r)$can be obtained with

$$\begin{aligned}
\phi_{d,0} &= (1-r)_+^l \\
\phi_{d,1} &= (1-r)_+^{l+1}[(l+1)+1] \\
\phi_{d,2} &= (1-r)_+^{l+2}[(l^2+4l+3)r^2+(3l+6)r+3] \\
\phi_{d,3} &= (1-r)_+^{l+3}[(l^3+l^2+23l+15)r^3+(6l^2+36l+45)r^2+(15l+45)r+15]
\end{aligned} \tag{5.83}$$

with $(1-r)_+^l$ being the cut off function such as

$$(1-r)_+^l = \begin{cases} (1-r)^l & if \quad 1-r \geq 0, \\ 0 & if \quad 1-r < 0 \end{cases} \tag{5.84}$$

These functions are defined to have compact support over the interval [0,1] and so, for our application we use scaling parameter θ to scale the function over the interval $[0, \theta]$ by replacing r by

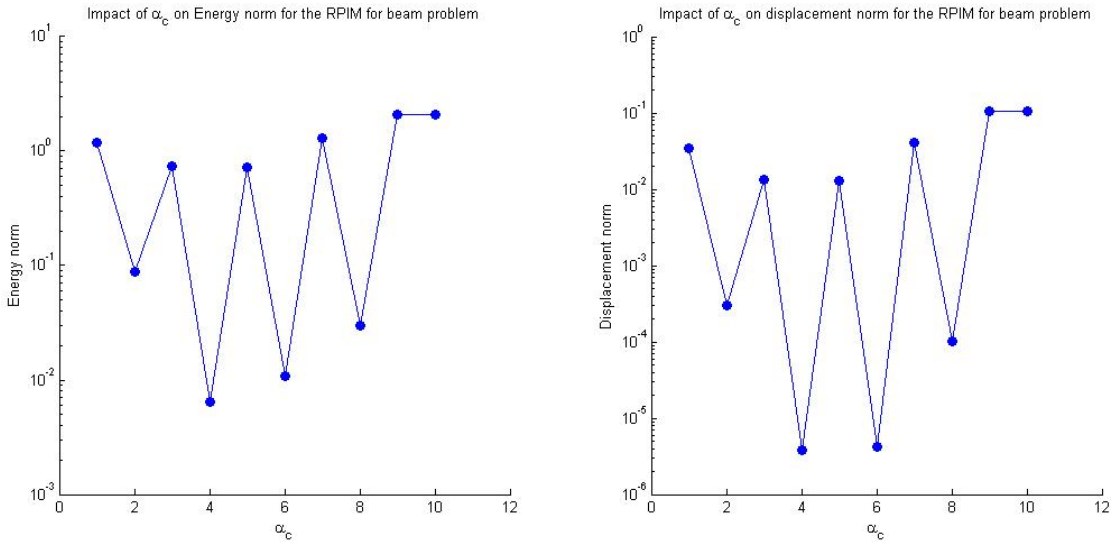


Figure 5.27: Impact of α_c on the energy norm and displacement norm for the RPIM TPS. 21×10 field nodes. $\alpha_c = 4.0$

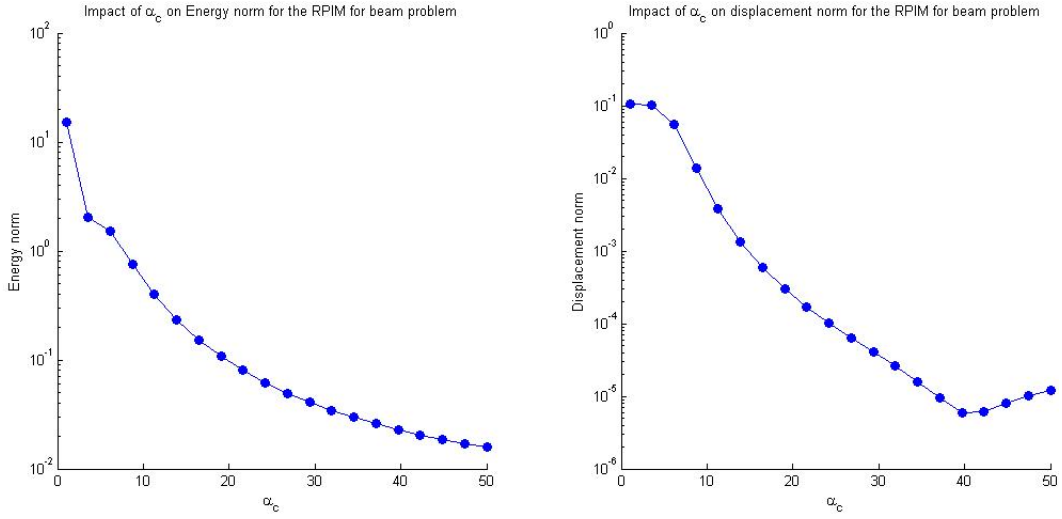


Figure 5.28: Impact of α_c on the energy norm and displacement norm for the RPIM WEN-C4. 21×10 field nodes. $\alpha_c = 2.0$

r/θ for $\theta > 0$. Scaling factors can vary through time and space but a protocol to specify optimal values for θ is yet to be defined. For these reasons, similar to normal RBF, empirical tests must be carried out to define suitable values for parameters over different types of problems. Wu's CSRBF are very similar to the Wendland but the process to obtain the functions is different.

For Buhman CSRBFs, the construction is different and he claims that it is a more general class of CSRBF that encompasses Wu and Wendland functions. We test the following Buhman functions which are C^2 and C^3 [50] respectively.

$$Buh - C^2 = \phi_{Bu,2} = \frac{1}{6} - 2r^2 + \frac{16}{3}r^3 - \frac{7}{2}r^4 + 2r^4 log(r) \tag{5.85}$$

$$Buh - C^3 = \phi_{Bu,3} = \frac{1}{9} - \frac{14}{15}r^2 - 7r^4 - \frac{16}{3}r^{\frac{7}{2}} + \frac{112}{45}r^{\frac{9}{2}} \tag{5.86}$$

Figure 5.30 shows a comparison for the 3 class of CS-RBF.

Buhman functions seems to allow greater values for optimal parameter θ over a wider range as

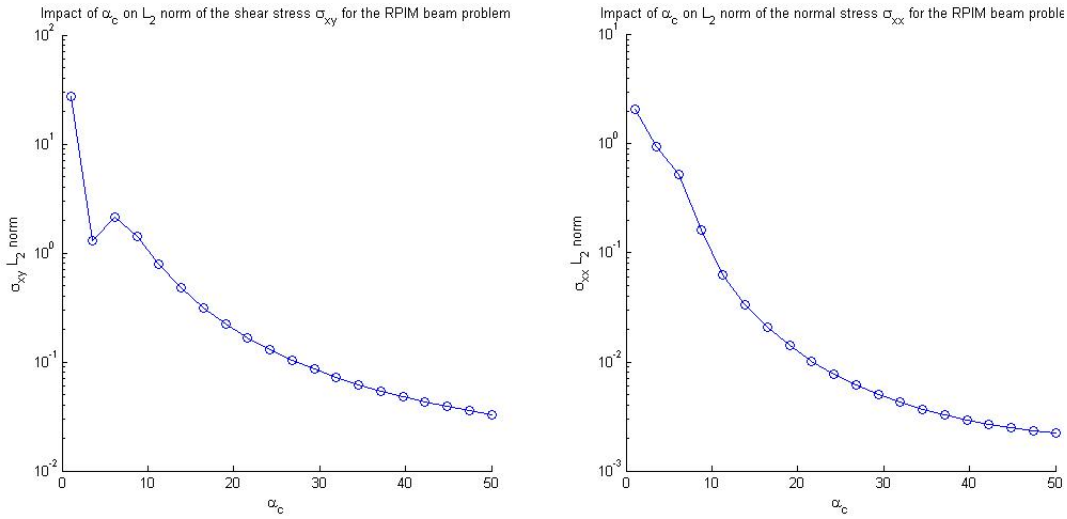


Figure 5.29: Impact of α_c on the L_2 error norm for the RPIM WEN-C4. 21×10 field nodes. $\alpha_c = 2.0$

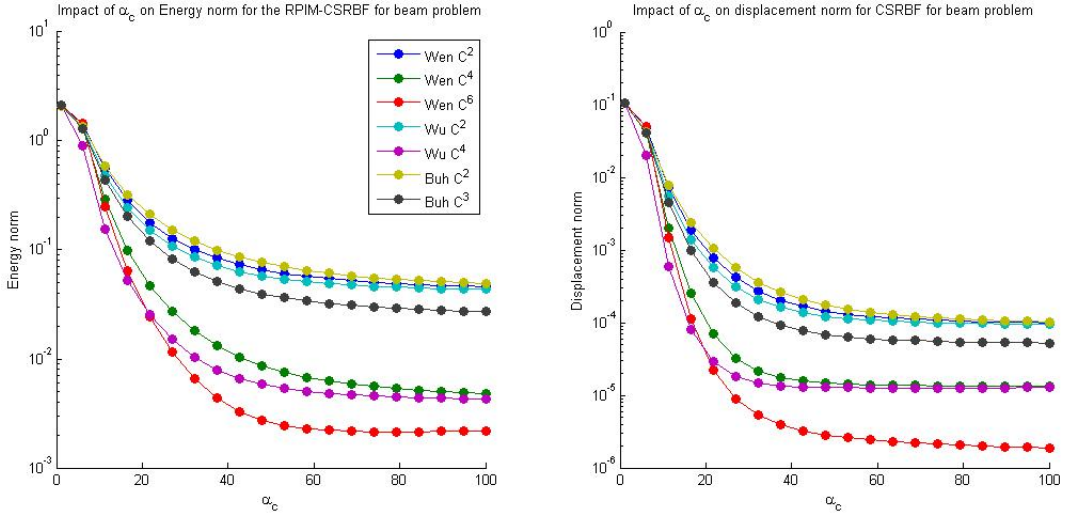


Figure 5.30: Impact of α_c on the energy norm and displacement norm for the RPIM-CSRBF. 21×10 field nodes. $\alpha_c = 3.0$

shown on figure 5.31 for the $Buh - C^3$. As expected the Wendland function, by providing a higher smoothness, yields the best results. They are widely used in image processing and seem to be a good choice for the RPIM. The advantage of CS-RBF is the compact support which will results in a sparse system matrix . To achieve this, the Euclidean norm r is normalized using the parameter θ. The parameter $\theta = \alpha_c \times d_c$ where d_c represents the average nodal distance surrounding node x_I at which the shape function is evaluated. It usually improves the computational efficiency for both time and storage.

Adding polynomial terms in the basis will usually improve the accuracy of RPIM and its stability. From the test, no drawbacks have been found.

These tests illustrate how important it is to use proper parameter's values. Having an insight on how the methods behave through the variation of these parameters is necessary to properly apply meshfree methods.

Integration is another sensitive process in the EFG. It is a costly procedure and significantly impacts the accuracy of the solution. In order to obtain the integrals in EFG, a background of cells

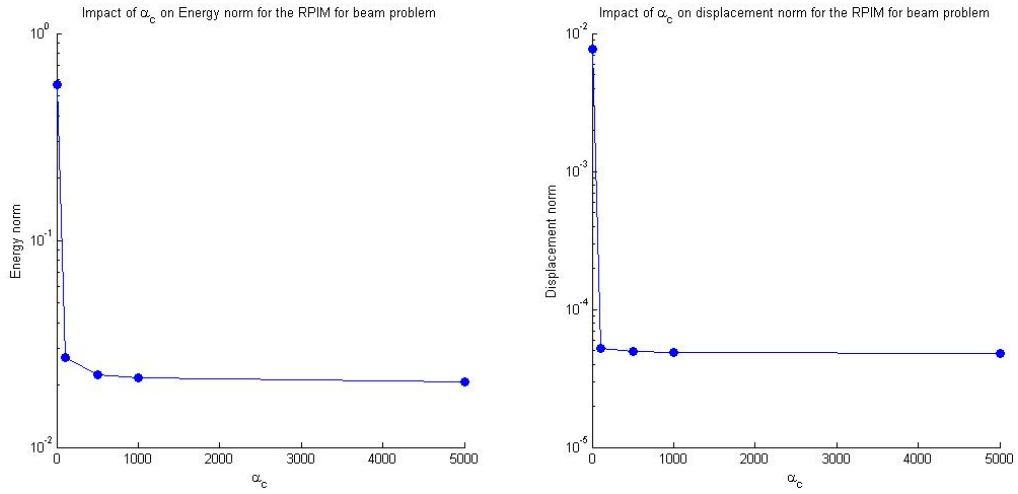


Figure 5.31: Impact of α_c on the energy norm and displacement norm for the RPIM Buh-C3. 21×10 field nodes. $\alpha_c = 3.0$

also referred as buckets [280] is used for the Gauss quadrature. The number of cells and Gauss point in each of them is directly related to the density of field nodes.

Integration has an important impact on the accuracy of the weak form based meshfree methods. In Global weak form methods, there are two parameters related to integration, i.e. the number of background cell and their topology, and the number of Gauss point within a cell.

We study the impact of the background cell refinement on the accuracy of the EFG. The number of Gauss point per rectangular background cell is set to 2×2. Figure 5.32 shows the impact of refining the background cell for different h of distributions of field nodes. Lagrange multipliers are used to enforce essential boundary conditions.

As the nodal field distribution is refined, the background cell should be refined to in order to maintain convergence of the method. For a fixed regular distribution of node 31×15, we study the impact of variation of the number of Gauss points per background cell. Figure 5.33 shows that increasing the number of Gauss points per background cell relax the requirement of the density of background cell in order to meet the convergence.

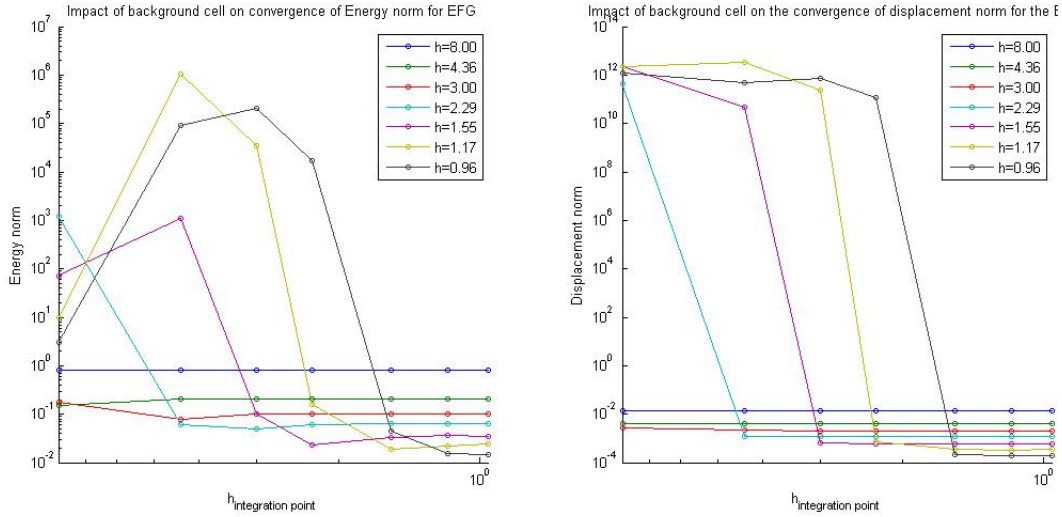


Figure 5.32: Impact of background cell on energy norm and displacement norm for EFG with MLS and cubic spline. 2×2 Gauss points per cell

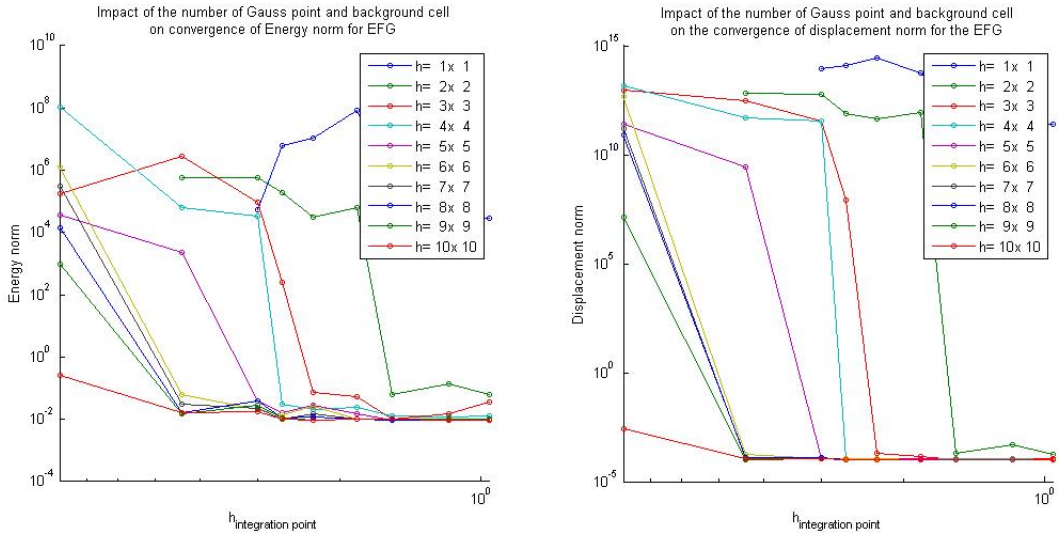


Figure 5.33: Impact of the number of gauss point per background cell on energy norm and displacement norm for EFG with MLS and cubic spline. Regular distribution of 31×15 field nodes

However, increasing the number of Gauss points per integration cell leads to an important increase in the computational cost of the method. The integration of MF SF derivatives due to their complex form is a source of error in meshfree methods.

5.6 Material discontinuity

Due to the high order continuity of meshfree SFs, material discontinuity cannot be dealt with naturally as it would be done in FEM where the mesh element is used to model material discontinuity at the material interface.

A set of nodes can be used as a discontinuity line to materialise the interface Γ_s between materials. These nodes belong at the same time to Γ^+ and Γ^-. The influence domain of nodes from each part of the interfaces cannot cross material interface and special treatment such as visibility criterion is required to build SFs. In that case weight functions and its derivatives exhibit discontinuity along the interface Γ_s. Constrained interface continuity condition has to be enforced to ensure approximated functions. Figure 5.34 illustrates a domain with material discontinuity where nodes from $\Omega+$ or $\Omega-$ only integrate nodes from their material domain in their respective influence domain.

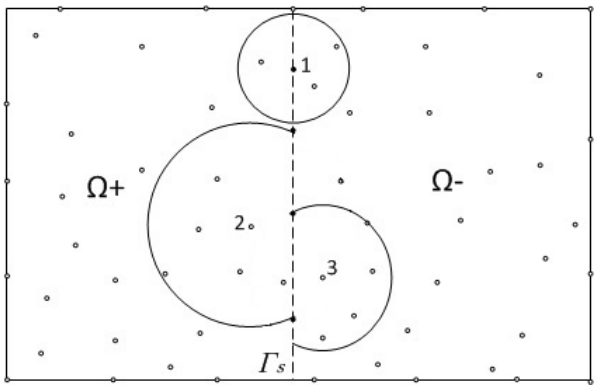


Figure 5.34: Treatment of material discontinuity and influence domain at interface

In practice, many MEMS are laminated structures built of many layers. Composite materials are

formed by bringing distinct materials together into close contact. The material properties of the assembly change discontinuously at the contact surface. Obviously, the mechanical conditions of the contact at the interface affect the rate at which energy flows across it. The analysis of heat transfer at a phase interface, requires specification of two conditions that must be added to the initial and boundary conditions of the problem formulation. The heat flux balance condition

$$(\nabla T \cdot \mathbf{n})_1 = (\nabla T \cdot \mathbf{n})_2 \tag{5.87}$$

$\mathbf{n}$ is the normal at the interface the two materials. The second condition on the relationship between the temperatures on both sides of the interface T_1 and T_2 is given by

$$(\nabla T \cdot \mathbf{n})_1 = h(T_1 - T_2) \tag{5.88}$$

With h the contact conductance. The perfect thermal contact condition states that at the interface $h \to \infty$ and $T_1 = T_2$.

5.7 Complex, non-convex boundary and domain singularities

In meshfree methods, the selection of the nodes involved in the SF computation has an important impact on the accuracy. For non-convex boundary problems or areas with discontinuities, a wrong selection of nodes can produce discontinuous SFs which will result in accuracy issues. The problem is to handle discontinuities or complex geometries singularities which lie in the influence domain of nodes. A solution is to use more nodes along the discontinuity and use a proper influence domain size as shown in section 5.3. To avoid the need for extensive nodal refinement, several methods have been developed to construct smooth weight functions around non-convex boundaries or singularity.

The visibility method [30, 280] is the simpler technique to treat discontinuities. It uses a ray tracing approach. An imaginary ray is considered between a point of interest within the domain of influence and the node. If the ray cuts a line of discontinuity within the influence domain, the point is discarded for building the weight function. Figure 5.35 shows a weight function domain modified using the visibility criterion. The issue is the discontinuity appearing along the segment from the discontinuity tip. Problems also arise for nodes close to the discontinuity line. Another

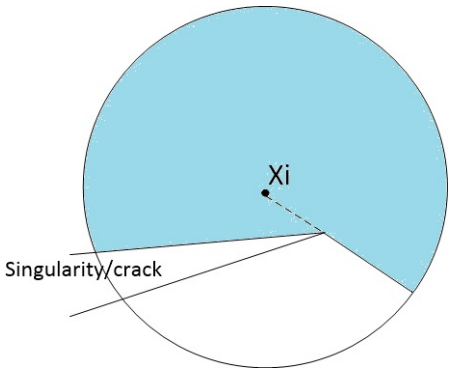


Figure 5.35: Weight function domain modified by the visibility method

very similar method is the see through method [281].

The transparency method [30, 211], allows some degree of transparency to the ray according the

distance from the point to the discontinuity boundary (crack or tip). Figure5.36 illustrates the transparency method for a node x_I in the vicinity of a singularity. The modified weight s(x) for

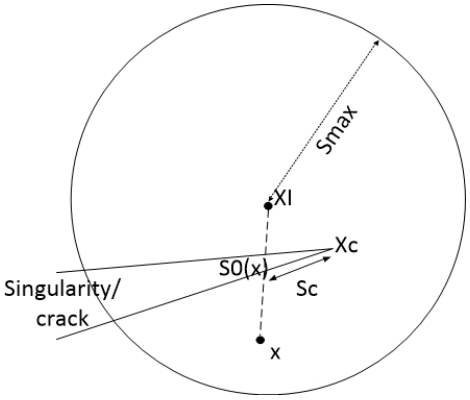


Figure 5.36: Transparency method

point x such that $[X_I, x]$ crosses the discontinuity is given by

$$s(x) = s_0(x) + s_{max}\left(\frac{s_c(x)}{s_n}\right)^{\lambda}, \quad \lambda \geq 2 \tag{5.89}$$

s_n is the distance at which the discontinuity line is completely opaque. Nodes should also respect a minimum distance from the crack surface. This prevents a scenario whereby the angle enclosed by the crack line and the ray from the node to the crack tip is too small, which in turn will lead to a sharp gradient in the weight function.

The diffraction method [211] is a well-accepted and accurate method. The nodal distance d_I is modified for x position for which the line $[x_I, x]$ intersects the singularity. Figure5.37 represents the typical diffraction modification.

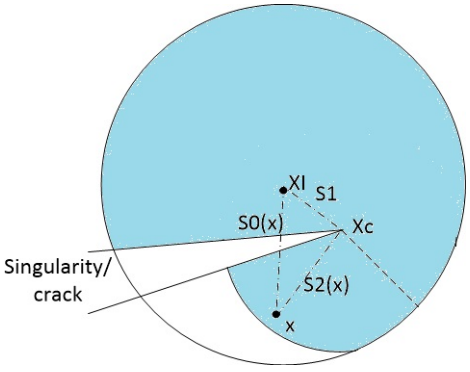


Figure 5.37: Weight function domain modified by the diffraction method incorporating singularity

The modified form of the nodal distance is given by

$$d_I = \left(\frac{s_1 + s_2(x)}{s_0(x)}\right)^{\lambda} \cdot s_0(x) \tag{5.90}$$

Where $s_1 = |x_I - x_c|$, $s_2 = |x - x_c|$, $s_0 = |x - x_I|$, with x_I the coordinate of the node considered, x the sampling point coordinate and x_c the coordinates of the crack tip. λ is called the diffraction parameter and is generally chosen to be between 1-2. Figure5.38 shows a weight function built around a crack singularity with $\lambda = 2$.

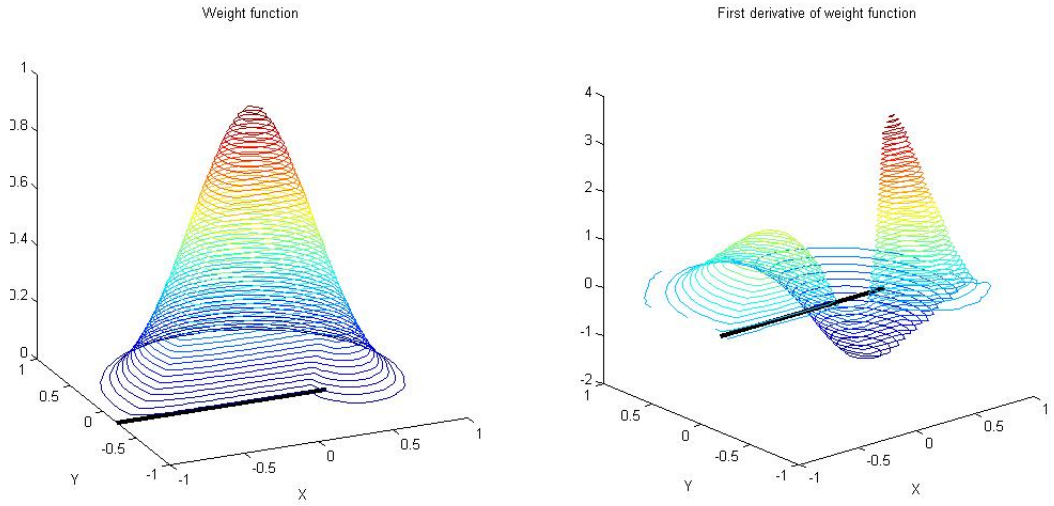


Figure 5.38: Quartic weight function with diffraction method

To limit steep variations of the weight function and its derivatives, the minimum distance from the nodes to the discontinuity line is used [211]. The diffraction method is more complex for 3D problems compared to the transparency method.

These techniques are commonly found in literature and have been applied for many cracks problems. However they are limited to domains containing one or two cracks in the support domain. They are not well suited for domains containing multiple cracks/singularities or non-convex boundaries. For this purpose, extension of the diffraction method [197] called the advance diffraction method and the relay method [159] have been proposed and implemented with success. However, they are more complex and will increase the computation cost of the weight function building process.

5.8 Conclusion

In this chapter we have reviewed meshfree methods based on weak formulations. The first implementation test of MLS and RPIM through EFG highlight the main features of meshfree methods as well as some difficulties in their implementation inherent to the methods. In general, the higher continuity of the solution provided by MLS or RPIM SFs allows for greater accuracy and rate of convergence than the standard FEM. Through the implementation benchmark in this chapter, it was found that wrong parameters for MF SFs can impact the conditioning for both the SFs moment matrix and the final system equation. This will lead to poor accuracy or even non convergence. With optimal parameters, EFG using both MLS and RPIM perform better than FEM and are also stable. They also performed better for irregular distribution of nodes compared to the mesh based techniques.

At first it seems that the mesh provides fastest computation whereas the meshfree techniques provide a more adaptive solution and more accurate result than FEM. In certain cases such as 3D problems, it is evident that the burden generated by the computation of the moment matrix in the process of building the SF at each node will have a serious impact on the computation cost.

All these data fitting approaches do not depend directly upon a mesh or any fixed relation between nodes. However, the implementation of such a method is not so simple in practice. There are often problems with stability and consistency. Many SF's parameters such as the support domain

size or optimal number of nodes in support domain, need to be determined to guarantee optimal fitting accuracy. Furthermore, in a Galerkin method, the discretisation of the differential operator, i.e., the integration of the stiffness matrix entries is in general quite involved in comparison with the conventional grid-based approach. This is especially the case for MLPG where the need to find intersections of the subdomain with boundaries makes the methods more computationally demanding. Another challenging task is the discrete formulation of Dirichlet boundary conditions, since the constructed SFs are not always interpolatory. Overall, Galerkin meshfree methods are computationally more demanding than the standard FEM due to the computational effort dedicated to building complex SFs and also the numerical quadrature.

Following our test, the advantages of local weak formulation based methods are not so clear. The implementation procedure for the system equation is very similar to the one for strong-form based methods such as FDM, where the assembly is simpler. The local weak formulation also removes the need for the SFs to be globally compatible.

However, the absence of a background cell for the integration results in a more costly integration process. This is particularly the case for nodes near boundaries where the intersection between the subdomain Ω_s and the boundary domain Γ needs to be properly computed by an efficient algorithm to ensure accuracy. This introduces another drawback associated with the local weak form method which is the increase in the data structure associated with the nodes. Basically each node carries its support and quadrature domain structures which encompass the list of nodes in the influence domain, quadrature points, and data related to the intersection with global boundaries. Depending on the geometry of the domain, this can become quite substantial.

MMs have a lot to offer through their intrinsic features such as their adaptive constellation of nodes and also higher order continuity and consistency shape functions compared to FEM. In the next chapter we apply global weak form methods though a series of benchmark problems to further study the performance of the method for thermal and elastic problems.

Chapter 6 Application and tests of meshfree methods based on weak formulation

6.1 Potential problem: two dimensional Poisson problem

Potential problems are found in many applications in modern engineering such as electrostatics, magnetostatics, elasticity, steady state heat conduction and incompressible fluid flow. In stationary diffusion or wave processes, the equation also reduces to the Laplace equation. Thermal problems are of importance in MEMS fabrication and design because many of the fabrication procedures involve thermal processes [23, 140]. Thermal stress can lead to several failures like cracks, fractures and delamination problems which complicate the analysis and simulation. The study of thermal effects is important in MEMS design as the fluctuations in ambient and operating temperature have a drastic impact on strain measuring circuits [1].
The general weak form of the 2D heat conduction equation is given in appendix B.1 p.206. We consider the particular case of the 2D Laplace equation over a square domain $\Omega \subset \mathbb{R}^2$ with only homogeneous prescribed boundaries conditions. The procedure followed is similar that the one for Poisson equation with a source function null.

$$k\Delta T = 0, \forall x, y \in \partial\Omega \tag{6.1}$$

$$T = \overline{T}, \forall x, y \in \Gamma_1 \tag{6.2}$$

$$\frac{\partial T}{\partial n} = \overline{q} \cdot n, \forall x, y \in \Gamma_2 \tag{6.3}$$

$$K\nabla T \cdot n = g(x, y) - hT, \forall x, y \in \Gamma_3 \tag{6.4}$$

Where n is the outward normal vector to the domain boundaries. k and h are the thermal conductivity and the convection heat transfer coefficient of the medium respectively. $\bar{T}$ and $\bar{q}$ are the prescribed boundary temperature and heat flux. The convection heat transfer coefficient is h. The RPIM and EFG methods use a Galerkin formulation in order to obtain the discretised linear system of equations. The variational form for a problem without internal sources is derived using the WRM. The weak formulation with the natural boundaries conditions is obtained by taking the residual form of the Laplace equation as follows

$$\int_\Omega W k \nabla^2 T d\Omega = 0 \tag{6.5}$$

by integrating by part we obtain the weak formulation over Ω of the problem

$$-\int_\Omega k\nabla W \cdot \nabla T d\Omega + \int_\Gamma kW\nabla T \cdot n d\Gamma = 0 \tag{6.6}$$

Replacing equation 6.3 and 6.4 into 6.6

$$\int_{\Omega} k\nabla W \cdot \nabla T d\Omega - \int_{\Gamma_3} hWTd\Gamma = \int_{\Gamma_2} \overline{q} \cdot n\Gamma - \int_{\Gamma_3} Wgd\Gamma \tag{6.7}$$

The essential boundary conditions on Γ_1 are applied by dedicated methods after the discretised system of equation is assembled. We now consider a heat conduction problem over domain $\Omega = [0, l] \times [0, h]$which is represented by the 2D Laplace equation with mixed boundary conditions.

$$\nabla^2(T) = 0 \quad \forall (x, y) \in \Omega \tag{6.8}$$

Together with the following essential boundary conditions

$$T(x, 0) = 0; \tag{6.9}$$

$$T(0, y) = 0; \tag{6.10}$$

$$T(x, h) = 100 \sin(\frac{\pi x}{10}); \tag{6.11}$$

and the flux right edge of the rectangular domain Ω

$$\frac{\partial T(l, 0)}{\partial x} = 0 \tag{6.12}$$

The analytical solution for the temperature scalar field is given by

$$T(x, y) = \frac{100 \sin(\frac{\pi x}{10}) * \sinh(\frac{\pi y}{10})}{\sinh(\pi)} \tag{6.13}$$

Figure 6.1 shows the temperature distribution obtained using EFG for a regular distribution of nodes. In our test to assess the performance of the MMs, we use the L_2 relative error norm defined

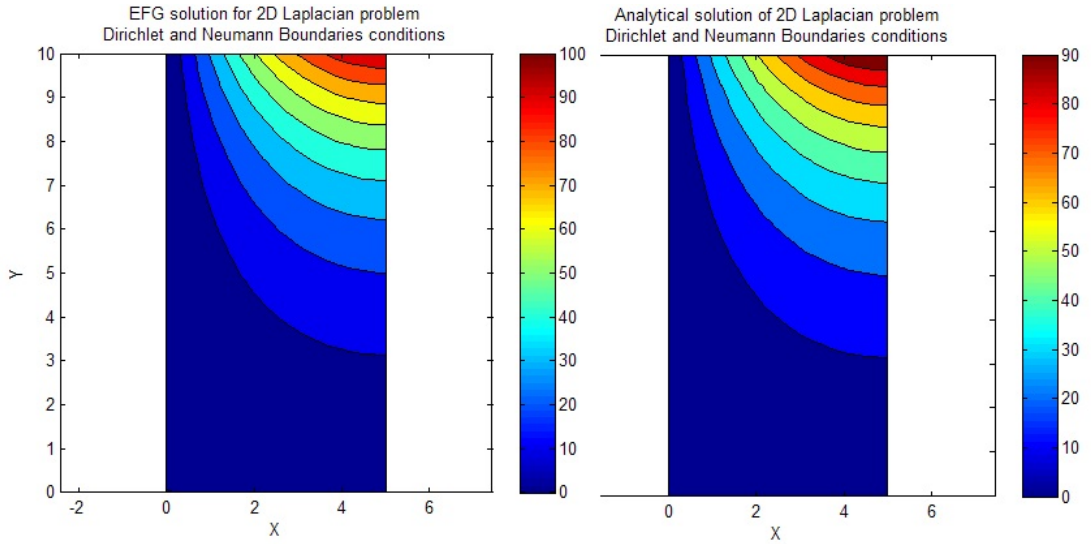


Figure 6.1: Temperature distribution obtained with EFG and MLS with a 19x19 field nodes regular distribution (left) vs Analytical solution(right). h = 5 mm,l = 10 mm

as

$$e_{L^2} = \frac{\|u^h - u_e\|_{L^2(\Omega)}}{\|u_e\|_{L^2(\Omega)}} = \left(\frac{\int_\Omega |u^h - u|^2 d\Omega}{\int_\Omega |u|^2 d\Omega} \right)^{\frac{1}{2}} \tag{6.14}$$

With u^h and u_e respectively the approximated solution and the exact solution. We study the impact of h refinement of the regular nodal distribution on the error norm. Figure 6.2 shows the convergence results for EFG with different weight functions for the MLS approximant and the size of the monomial basis m=3.

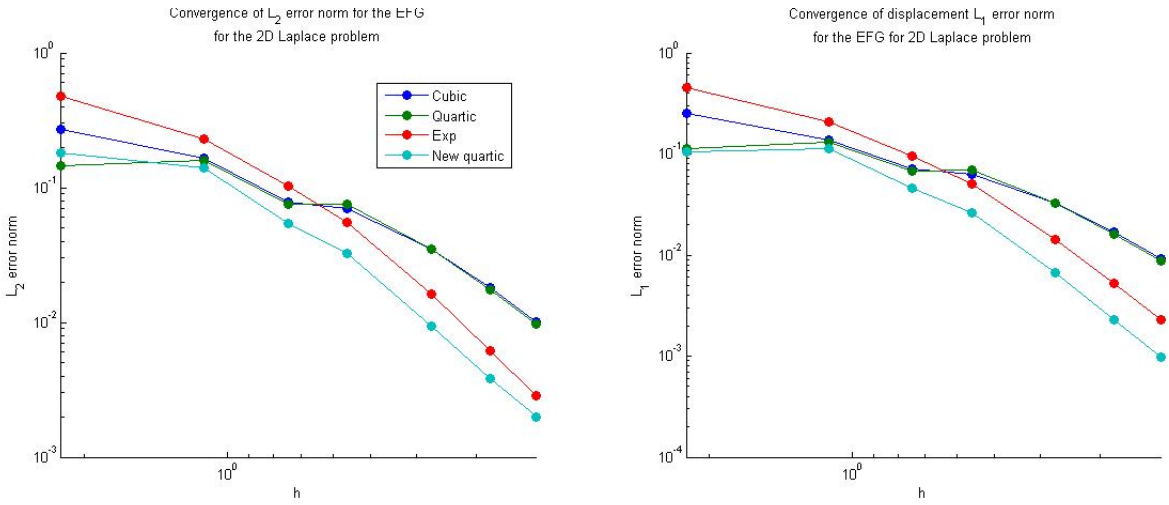


Figure 6.2: Convergence of the L_2 and L_1 relative error norm for the 2D Laplace problem with EFG. $\alpha_s = 3$

Figure 6.3 represents the increase of computational time (CPU) as the nodal density is refined. The figure also compares the convergence rate (R) with the increase rate of CPU time (R_{CPU}) for each SFs. This is represented on the right diagram of figure 6.3.

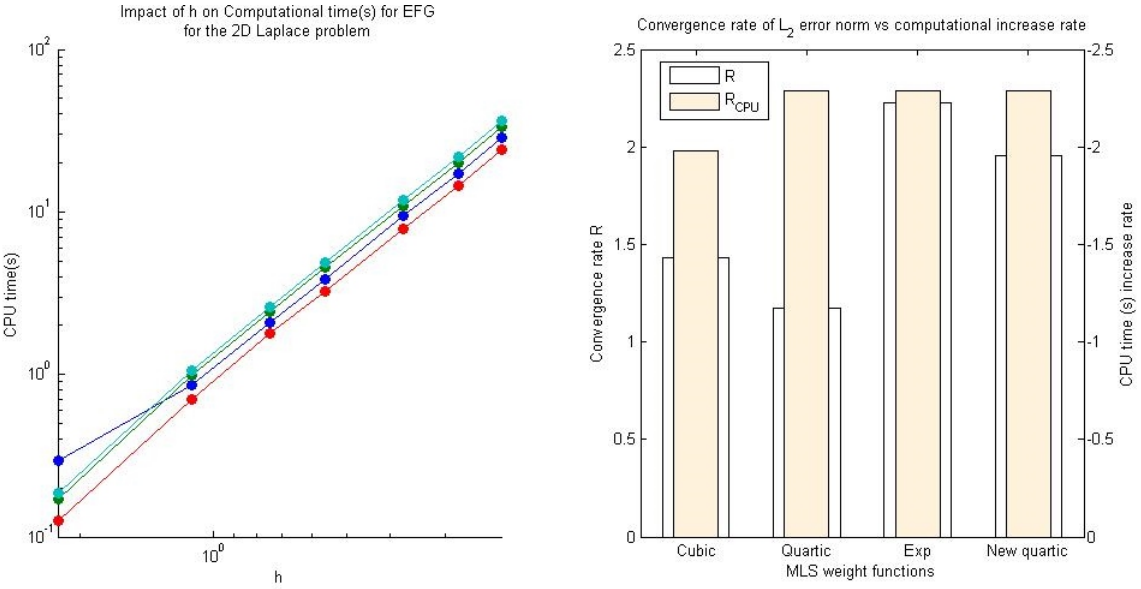


Figure 6.3: Impact of h refinement on the computational cost (left) and comparison of convergence rate with computation cost increase rate(right).

As we can observe, all the SFs provide quasi-similar results in term of increase rate of computational cost. However the MLS with cubic spline has the best ration of R/R_{CPU}.

By increasing the monomial basis order, we obtain a much better error with an increased rate of convergence as shown on figure 6.4 and 6.5 where m=10. Figure 6.6 shows the impact of increasing the support size α_s for various monomial basis size m for the MLS with exponential weight function(Exp).

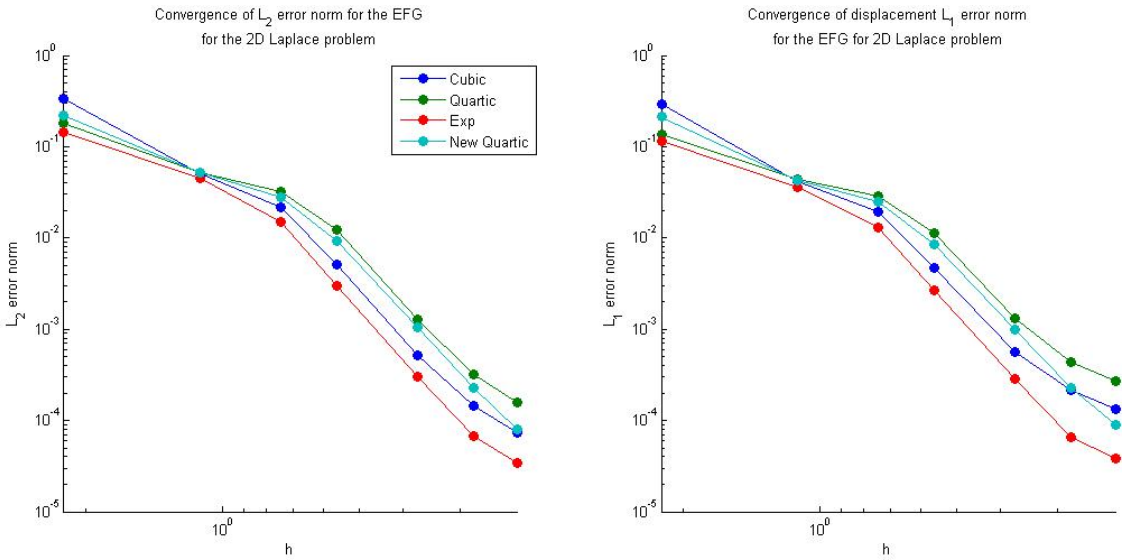


Figure 6.4: Convergence of the L_2(righ) and L_1 (left) relative error norm for the 2D Laplace problem with EFG. $\alpha_s = 3$

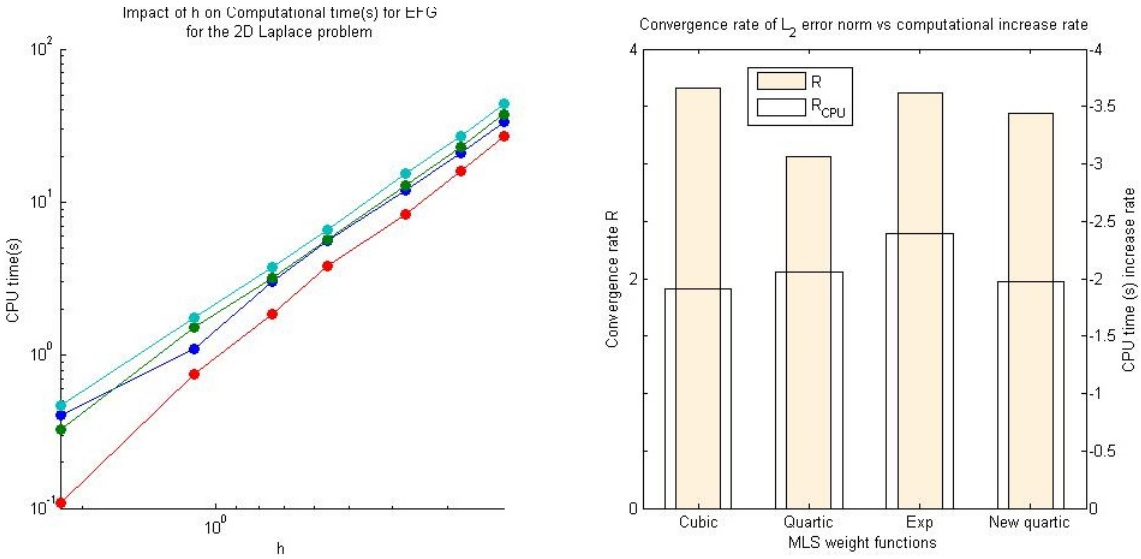


Figure 6.5: Impact of h refinement on the computational cost and comparison of convergence rate with computation increase rate.

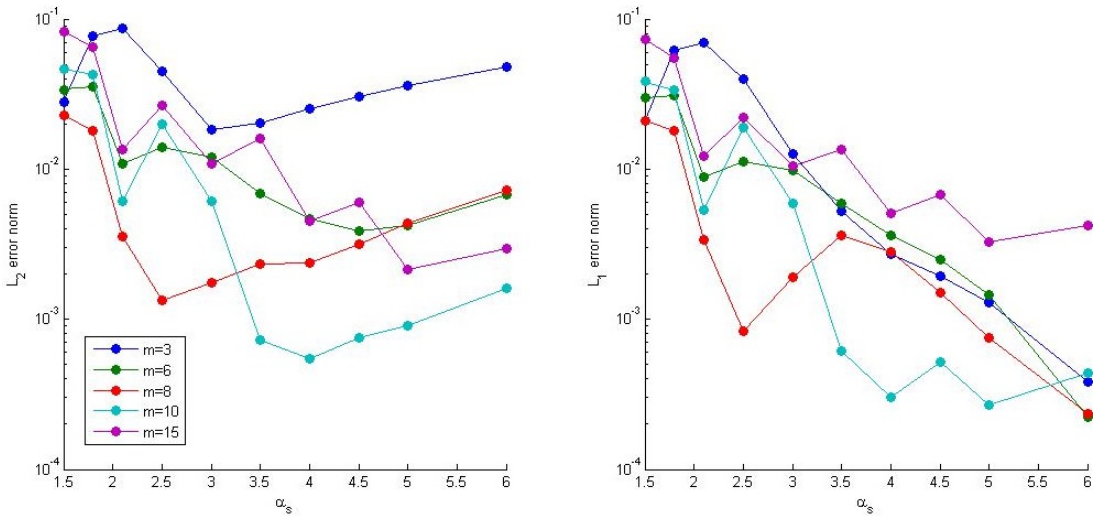


Figure 6.6: Impact of α_s on L_2 error norm for different basis size m. 4×4 Gauss point per cell.

Figure 6.6 shows that increasing m along with the support size domain through the dimensionless parameter α_s provides a similar effect as the p-refinement in FEM.
Increasing α_s improves the accuracy. However when the support domain is too big, the accuracy degrades for the L_2 error but not the L_1. Increasing the basis size m improves the accuracy. However as the order of monomial basis increases, more nodes are needed in the support domain to fulfil the rank of the MLS moment matrix to ensure that it is invertible. Thus as shown on figure 6.6, an increase of α_s is needed to take the most out of the increase of the monomial basis order. Integration becomes harder and more Gauss points should be used to obtain optimum accuracy. In our case we use 4×4 Gauss point per background cell.
Figure 6.7 shows that increasing the basis does not significantly impact the computational cost of the method and provides a convergence similar to the h convergence previously studied when refining the nodes density.

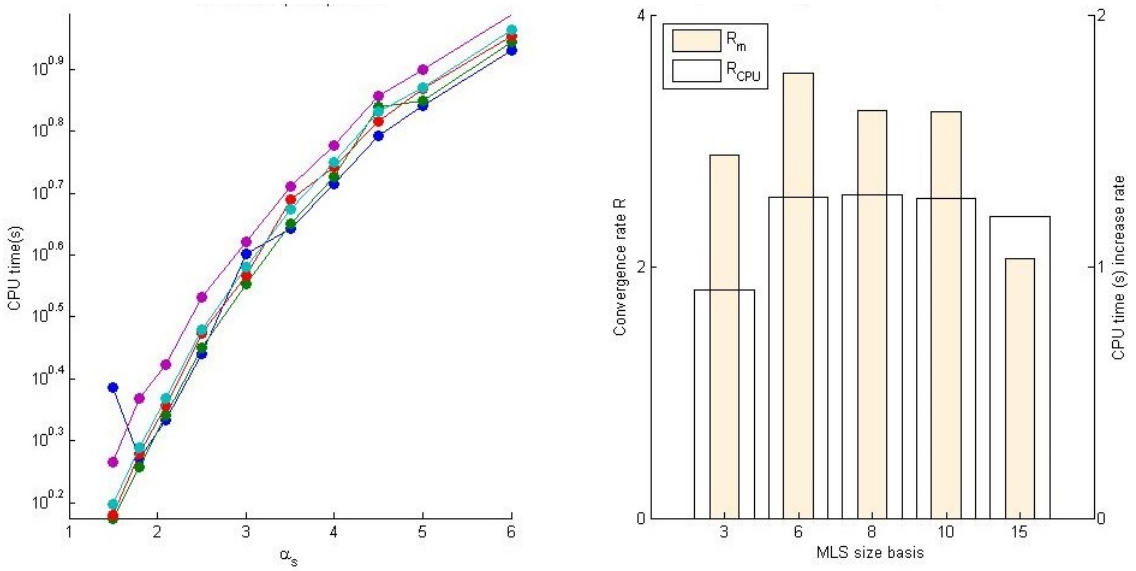


Figure 6.7: Effect of α_s on the computational time(left) and comparison of convergence rate with increase of computational rate for MLS with different basis size.

When the support domain is too small, there are not enough nodes within it to properly build the MLS interpolant due to rank deficiency when trying to invert the moment matrix.

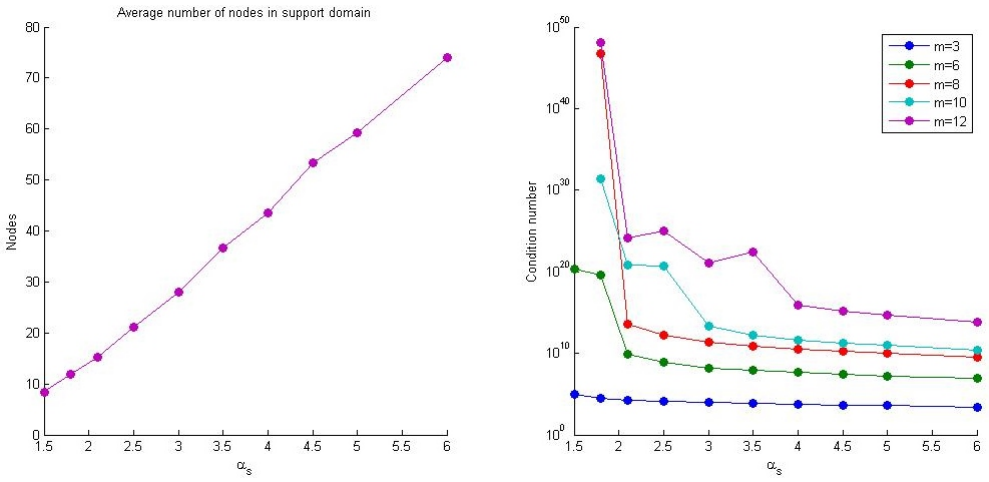


Figure 6.8: Impact of α_s on the condition number of the MLS moment matrix for different basis size(right). Average number of nodes in support domain for different α_s (left).

As shown in figure 6.8, when not enough nodes are used, it results in a badly conditioned moment matrix. This requires special treatment to compute the shape function. In our case we use the pseudo inverse for $\alpha_s < 2$ for m= 8, 10, 15 which leads to good results as shown on figure 6.6. This results in instabilities and increase in the computational cost for the building of the shape function. This situation should be avoided by a simple technique such as adding nodes in the area.

We now study the convergence of the RPIM for different RBF and CRBF. Figure 6.9 shows rather similar results as those obtained by the EFG with MLS interpolant.

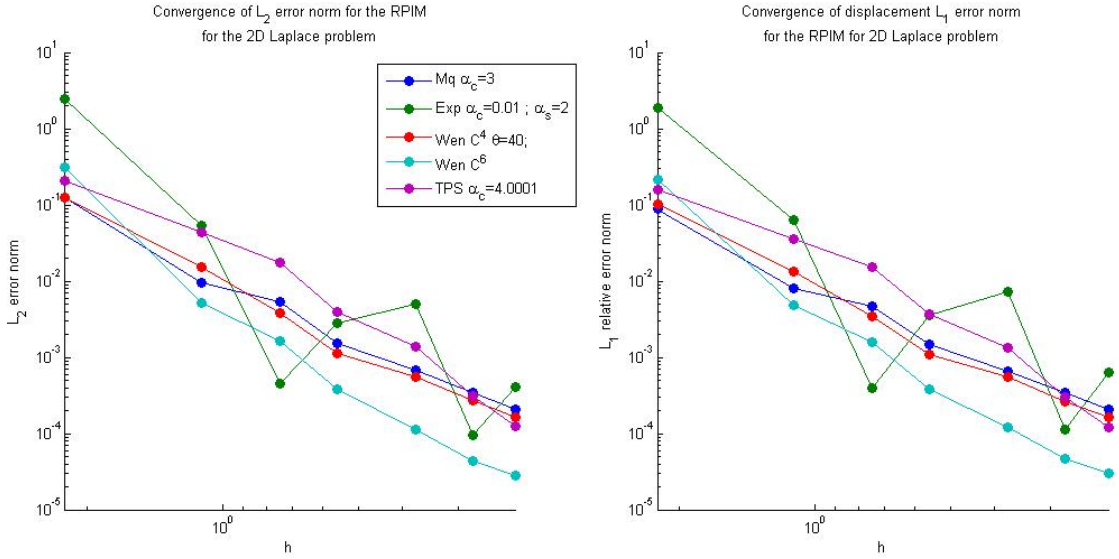


Figure 6.9: Convergence of the L_2(left) and L_1(righ) relative error norm for the 2D Laplace problem with RPIM.$\alpha_s = 3$

Figure 6.10 shows the convergence obtained different RPIM methods for irregular distribution of nodes. Figure 6.11 compare the Rate of convergence with increase in computational rate (R_{CPU}). We observe on figure 6.10 that RPIM has rate of convergence much important for the first the first refinement step. On the right of figure 6.11, we differentiate the first part of the convergence rate (transparent bar) and second part (blue) which is much less. Overall no benefit is found using CSRBF over RBF for heat conduction problem.

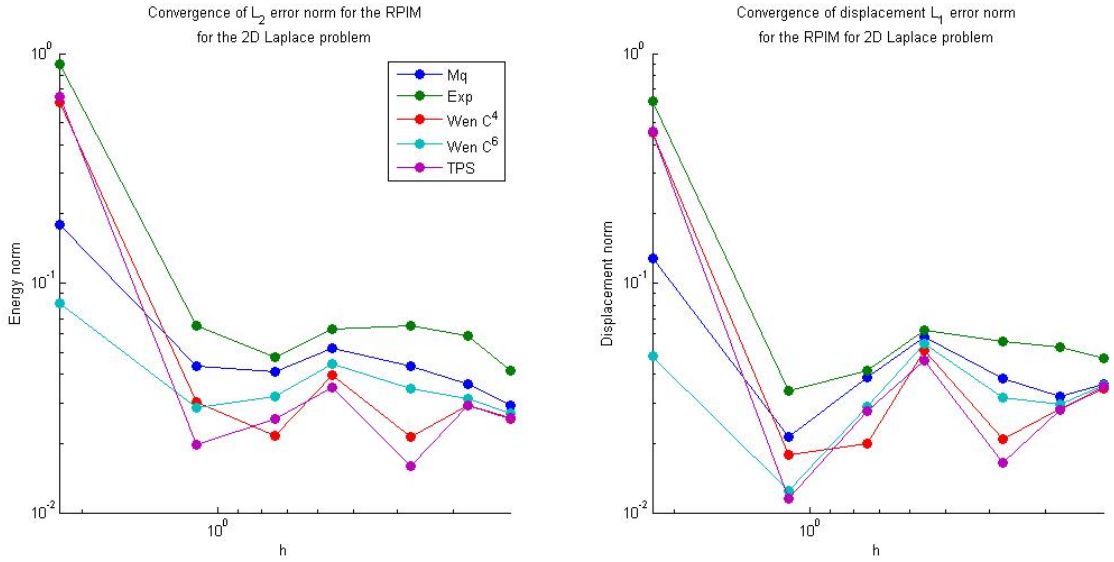


Figure 6.10: Convergence of the L_2 and L_1 relative error norm for the 2D Laplace problem with RPIM for irregular node distribution.$\alpha_s = 3$

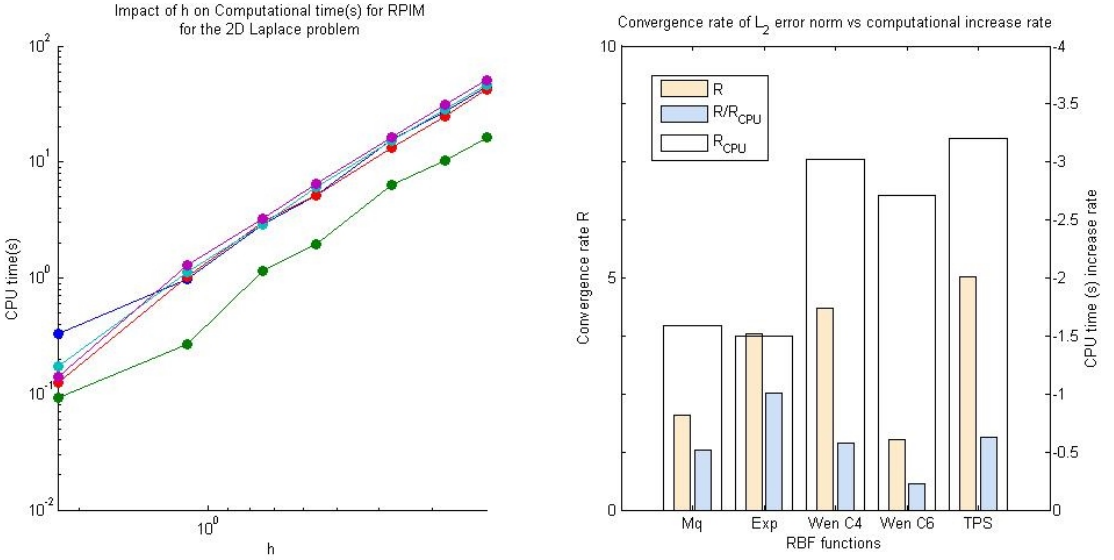


Figure 6.11: Impact of h refinement on the computational cost for RPIM and comparison of convergence rate(beige) with the computation increase rate. Irregular nodal distribution

We now compare the EFG with an FEM developed using Quad element. Figure 6.12 shows a comparison of the convergence of RPIM , EFG and FEM. Figure 6.12 illustrates the capability of RPIM and even more so of MLS to exhibit a higher rate of convergence when a higher basis is chosen. However, when looking at the computational cost it appears that FEM is less costly than both EFG with MLS and RPIM.

The support domain is a sensitive parameter in EFG. It controls the number of nodes used to build the SFs. When the number of nodes is too important, the SF is too smooth and the computational cost increases drastically especially for the computation process of SFs functions. In the case where it is too small, there is a risk that the domain Ω will not meet the patch cover criteria for convergence in the global Galerkin weak form framework. It will also produce incorrect shape functions. The domain of influence is conditioned by the local density of field nodes in the region. For an optimal implementation of EFG, when there is no prior knowledge on the nodes density,

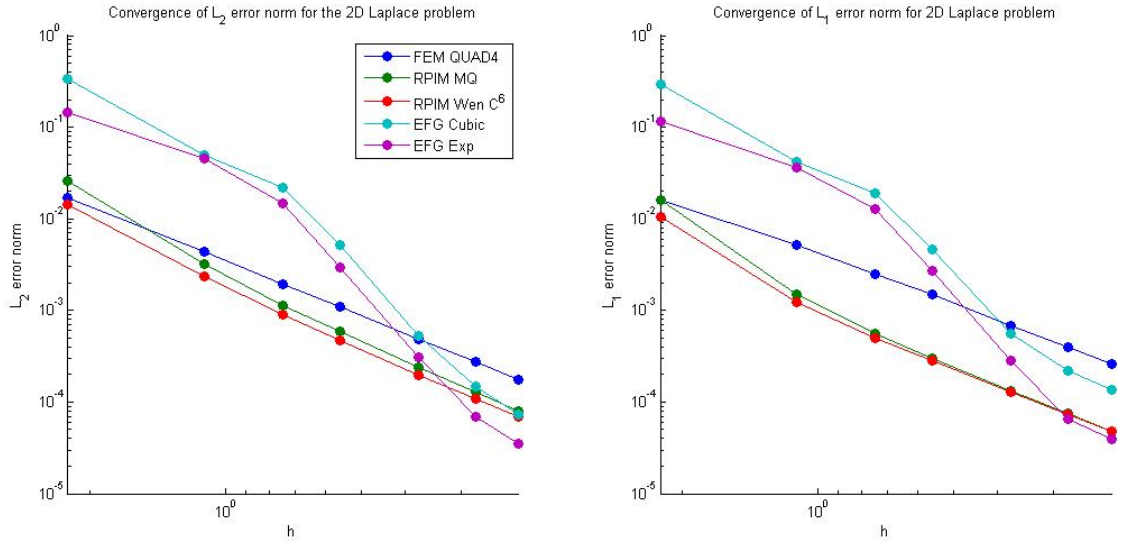


Figure 6.12: Comparison of FEM, EFG and RPIM convergence of the L_2(left) and L_1(right) relative error norm for the 2D Laplace. $\alpha_s = 3$. 2×2 Gauss points

an assessment process of the density should be performed to obtain the nodal density parameter. This will allow optimization of the size of the influence domain for better accuracy and faster computation. Figure 6.13 shows that FEM exhibits the smallest CPU time and convergence rate.

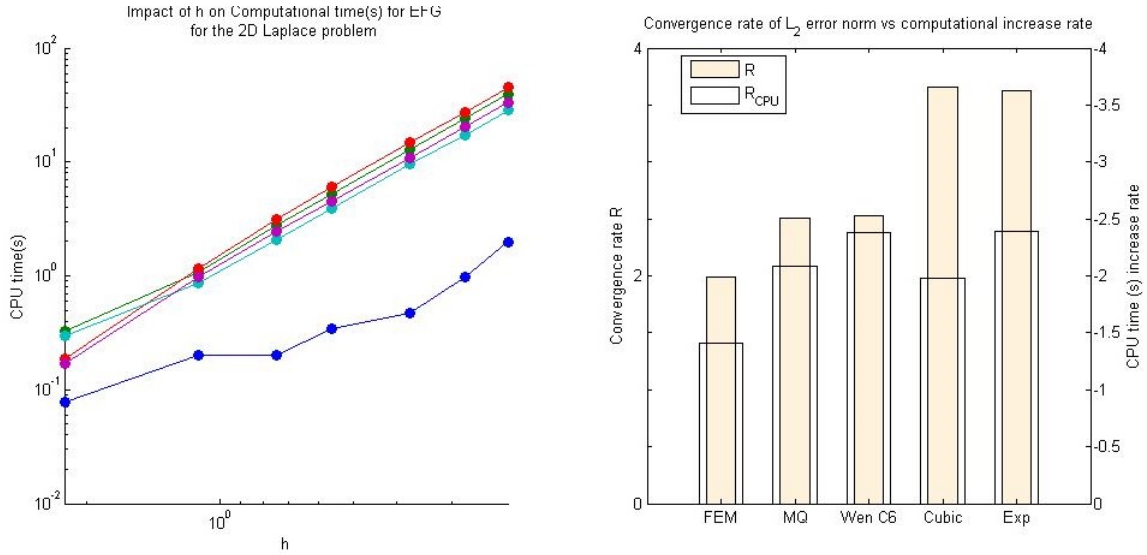


Figure 6.13: Impact of h refinement on the computational cost(s) and comparison of convergence rate with computation increase rate with irregular nodes distribution

This however, is for the case of simple geometry problems. In general, the computational time for FEM increases greatly along with the mesh complexity. On the other hand, for MMs, the CPU time rate does not increase much with the complexity of the domain due to the nodal feature. In this regard, the higher convergence rate of MMs is clearly an advantage over FEM.
We now study the same problem using the SNCI method (see section 5.1.4.4 p.103). Taking the weak formulation for the 2D dimensional heat conduction equation

$$\int_\Omega \delta(\nabla T)^T k \nabla T d\Omega - \int_\Omega \delta T^T f d\Omega + \int_{\Gamma_2} \delta T^T \bar{q} d\Gamma + \int_{\Gamma_3} \delta T^T h T d\Gamma - \int_{\Gamma_3} \delta T^T h T_a d\Gamma = 0 \quad (6.15)$$

Giving the following final matrix system

$$KT = F \tag{6.16}$$

where

$$K_{IJ} = \int_{\Omega} (\nabla \Phi_I)^T k \nabla \Phi_J d\Omega + \int_{\Gamma_3} h \Phi_I^T \Phi_J d\Gamma \tag{6.17}$$

and

$$F_I = \int_{\Omega} \Phi_I^T f d\Omega - \int_{\Gamma_2} \Phi^T \bar{q} d\Gamma + \int_{\Gamma_3} h \Phi_I^T T_a d\Gamma \tag{6.18}$$

The domain Ω is partitioned in k Ω_k smoothing domains for each field nodes such as $\Omega = \sum_{k=1}^{N} \Omega_k$. Using nodal integration we write

$$K_{IJ} = \sum_{k=1}^{N} K_{IJ}^{(k)} \tag{6.19}$$

with

$$K_{IJ} = \int_{\Omega_k} (\nabla \Phi_I)^T k \nabla \Phi_J d\Omega + \int_{\Gamma_{k_3}} h \Phi_I^T \Phi_J d\Gamma \tag{6.20}$$

Γ_k represent the boundary of the smoothing domain for node k. We apply the smoothing operation on the nodal field gradient [57, 58] over the cell associated with the node k such as

$$\frac{\partial \Phi_I(x_k)}{\partial x} = \int_{\Omega_k} \frac{\partial \Phi_I(x)}{\partial x} W(x - x_k) d\Omega \tag{6.21}$$

W is the smoothing function which is chosen as

$$W(x - x_k) = \begin{cases} \frac{1}{A_k} & \forall x \in \Omega_k \\ 0 & \forall x \notin \Omega_k \end{cases} \qquad A_k = \int_{\Omega_k} d\Omega \tag{6.22}$$

Here, A_k is simply the smoothing area of the cell Ω_k. Equation 6.21 becomes

$$\frac{\partial \Phi_I(x_k)}{\partial x} = \frac{1}{A_k} \int_{\Omega_k} \frac{\partial \Phi_I(x)}{\partial x} d\Omega = \frac{1}{A_k} \int_{\Gamma_k} \Phi_I(x) n_j d\Gamma \tag{6.23}$$

where n_j is the normal to the boundary segment γ_k. We can use Gauss integration along each segment of the boundary Γ_k of the smoothing domain Ω_k or a simple trapezoidal rule to reduce the computational cost. The integration cells are built using the mid point technique from a Delaunay triangulation. Figure 6.14 (left) shows the cells built over a Delaunay triangulation using our Matlab code.

Similar cells can be built over quad elements, however in our procedure we first build a triangulation out of the quads mesh, then the integration cell. Figure 6.14 (right) shows the cells obtained from such meshes. Building the cells from quad elements is not the best solution because when the nodal distribution is irregular, the triangulation produced is of poor quality.

Figure 6.15 shows the temperature field solution obtained with EFG using nodal cell integration(NI-EFG).

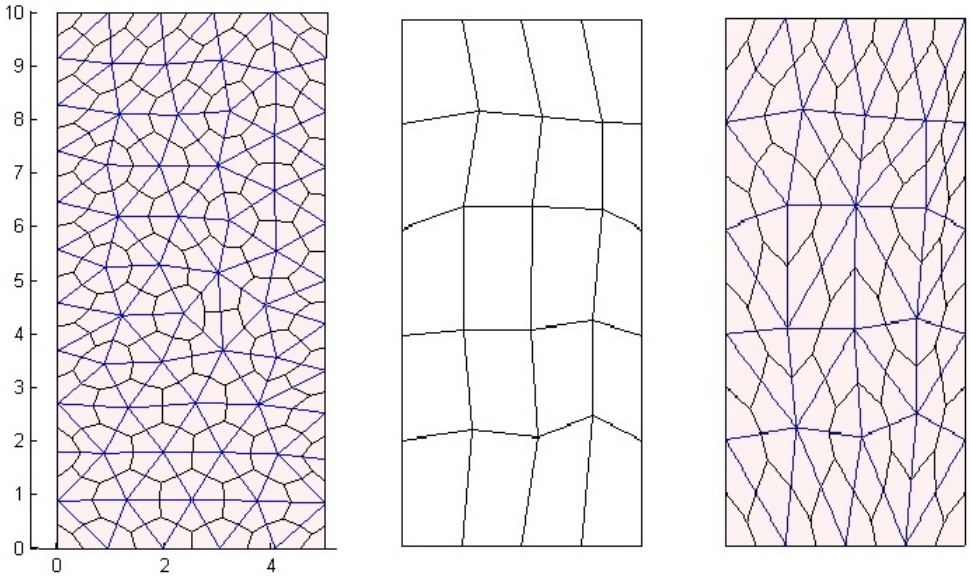


Figure 6.14: Integration cells (black) built over Delaunay triangulation (blue) of irregular 30 nodes (left). Quad mesh (middle) and cells built over the triangulation built from quad mesh (right)

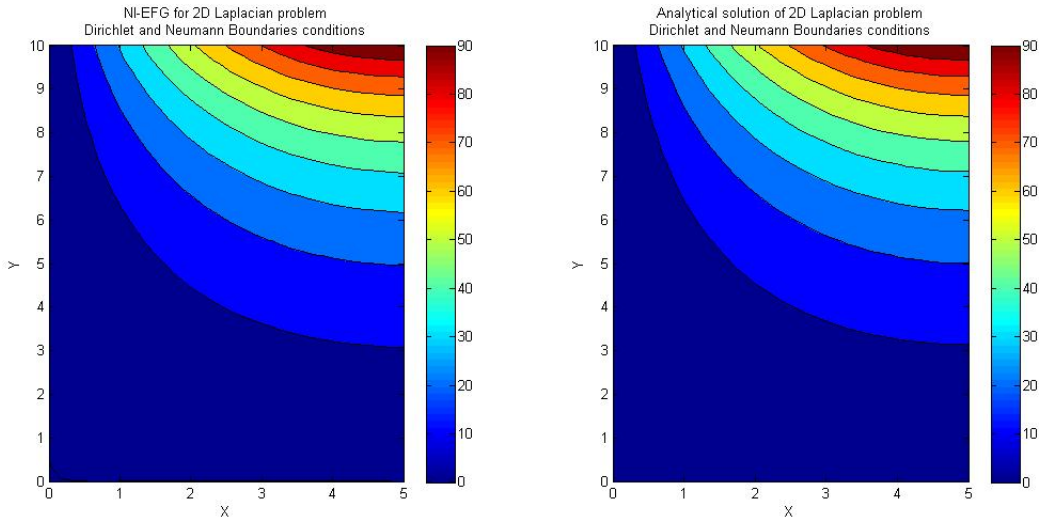


Figure 6.15: Temperature distribution obtained with NI-EFG and the analytical solution. Irregular distribution of 30 field nodes. L2 error= 6.0×10^{-3}.

Figure 6.16 shows the y-component of the gradient of the temperature field with $L_2 = 0.03$. Our test confirmed that better accuracy for the function field as well as for the gradient is obtained using a smoothing nodal integration. However the process of building the cell involves the generation of a triangulation (mesh) which is in contradiction with our motivation of prospecting for meshfree methods solution.

6.2 2D high gradient heat conduction

We consider the heat conduction problem [280] over a rectangular plate (0.5×6) with the heat source

$$b(x, y) = 2s^2 sech^2(s(y-3))tanh(s(y-3))\ W.m^{-2} \tag{6.24}$$

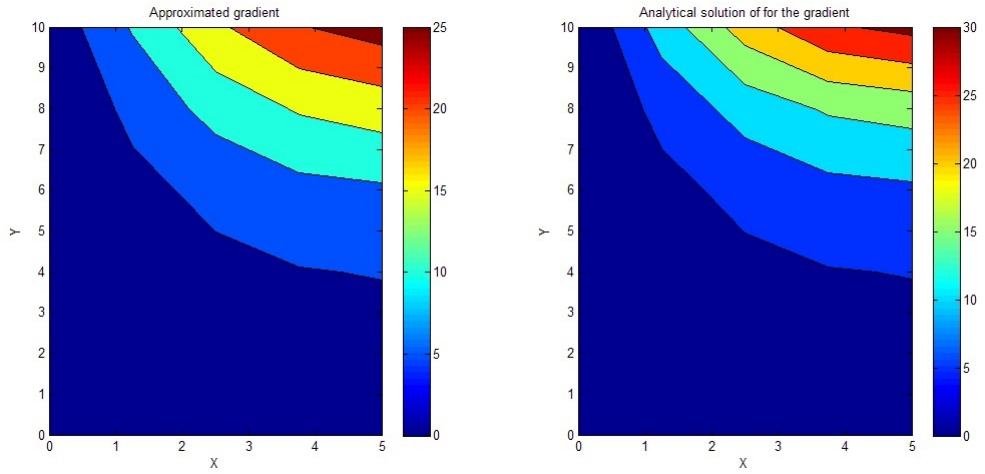


Figure 6.16: y component of the Temperature gradient obtained with NI-EFG and the analytical solution. Irregular distribution of 30 field nodes. L2 error= 3.0×10^{-2}.

and the following boundary conditions

$$T(x,0) = -tanh(3s)\,^{\circ}C, \quad and \quad T(x,6) = tanh(3s)\,^{\circ}C \tag{6.25}$$

$$\frac{\partial T(-0.25,y)}{\partial x} = 0\,^{\circ}C, \quad and \quad \frac{\partial T(0.25,y)}{\partial x} = 0\,W.m^{-2} \tag{6.26}$$

The weak formulation of such a problem for EFG is given in section B.1, page 206. This problem presents a high temperature gradient at the vicinity of y = 3. The parameter s controls the gradient of the temperature field at y = 3. The bigger s is, the higher the gradient. The challenge in the numerical method is catching the pattern of the gradient of the temperature in the vicinity of y=3.Such problems require the use of a very small mesh size h to capture the variation of the gradient using a linear shape function. We use the Penalty method to enforce the essential boundary condition with $\alpha_p = 10^8$. The general weak formulation of the problem is given in 6.44 where Q(x,t)= b(x,y) for the present case. Figure 6.17 and 6.18 shows the temperature field and gradient field obtained with meshfree EFG as compared to that obtained by the analytical solution for the parameter s = 50. With FEM, in obtaining the gradient of the approximated temperature,

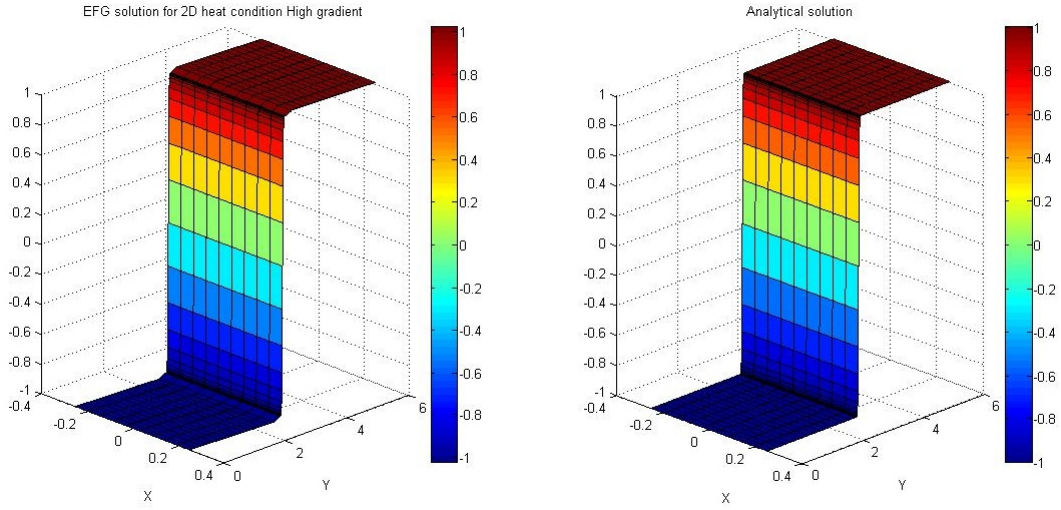


Figure 6.17: Comparison of the temperature field obtained with EFG (left) and exact solution (right). s=50 and $\alpha_s = 2.0$

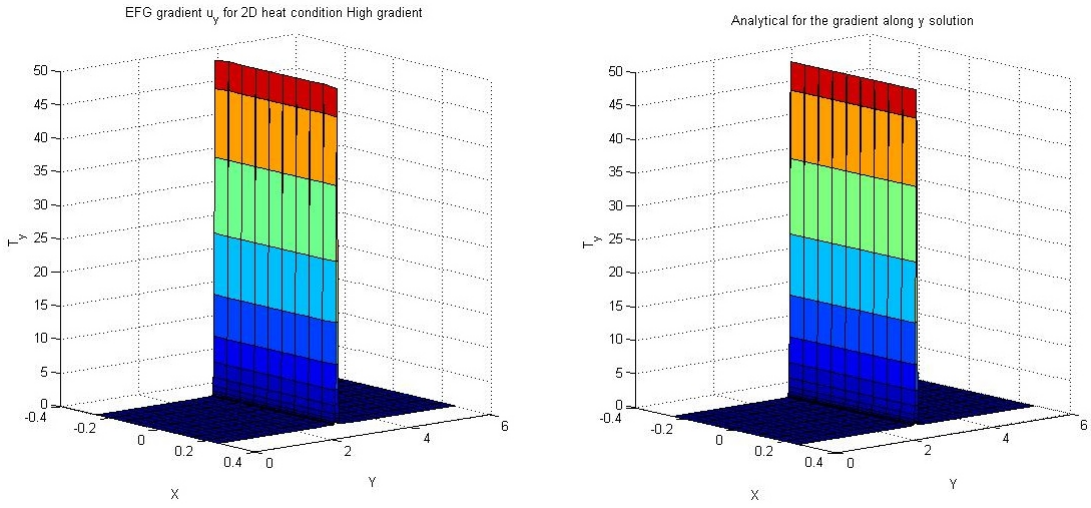


Figure 6.18: Comparison of the gradient temperature field obtained with EFG (left) and exact solution (right). s=50 and $\alpha_s = 2.0$

we encounter the same problems as obtaining stress and strain from primary unknowns. Within elements, the stress and in this case the temperature gradient is constant. Post-processing techniques are usually applied to provide a smooth gradient at precise locations within the elements. In meshfree, the approximation of the gradient at a particular nodal location is obtained directly from nodal approximation using the derivative of meshfree shape function. The resulting gradient is much smoother compared to FEM and no post-processing is required.

We set the parameter s = 10. Figure 6.19 shows that Galerkin meshfree methods using both MLS and RPIM shape function provide better accuracy compared to the regular FEM Q4 with 4×4 Gauss points. This becomes more apparent for the approximated gradient. Regular background cells with 2×2 Gauss points are used for the integration of the Galerkin form. To evaluate the gradient error, we use the equivalent gradient energy norm or equivalent energy

$$e_e = \sqrt{\int_\Omega (g^{exact} - g^{num}) \boldsymbol{k} (g^{exact} - g^{num}) d\Omega} \tag{6.27}$$

Where g is the temperature gradient. As the preliminary studies have demonstrated, RPIM SFs

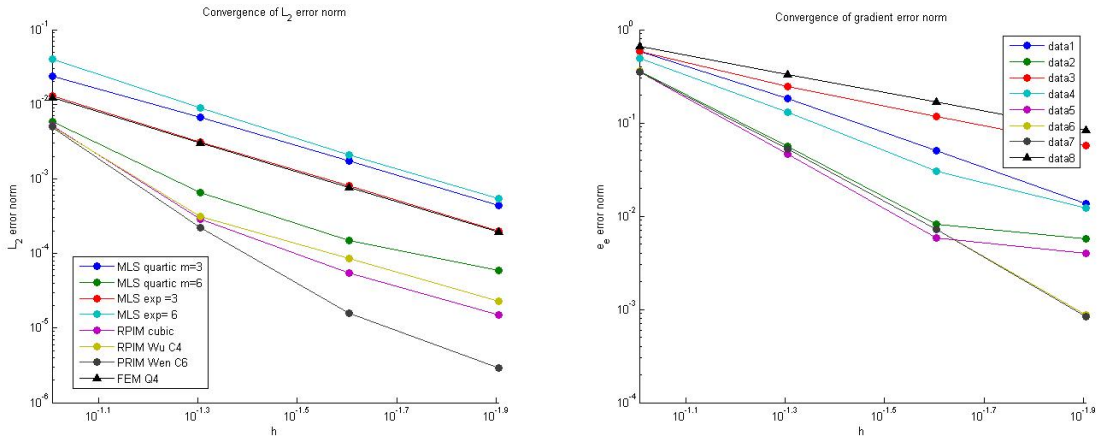


Figure 6.19: Convergence of L_2 error norm and gradient energy norm. s=10 and $\alpha_s = 2.0$

with higher continuity RBF provides the best accuracy. For MLS, higher basis size m such as quadratic also provides improved accuracy for such stiff problems. The convergence rate as shown

in figure 6.20 is also much higher for EFG. We now set s=50. Figure 6.21 shows the results

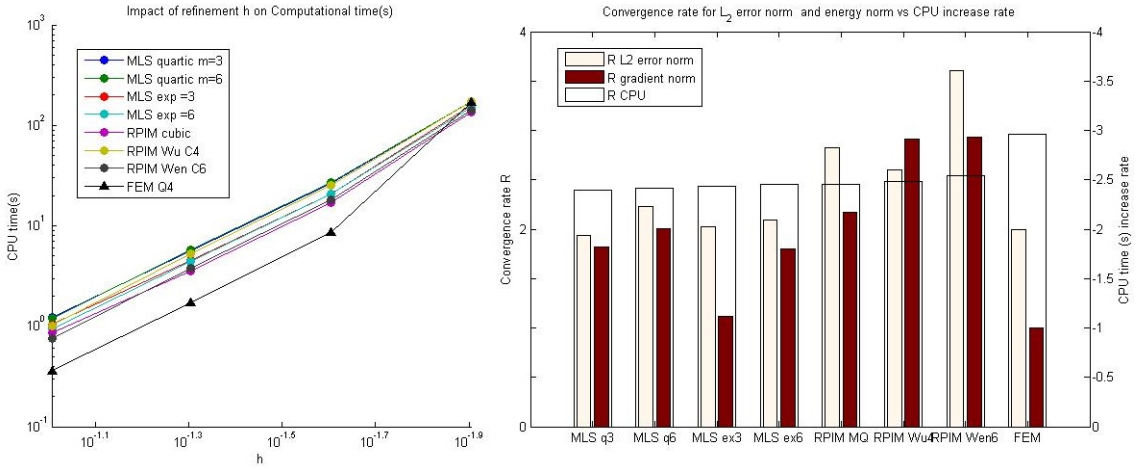


Figure 6.20: Comparison of convergence rate for the L_2 error norm, gradient norm and CPU cost

obtained, for the convergence of the L_2 norm. Figure 6.22 shows the influence on setting the s=50 on the Gradient L_2 error norm and the computational cost in second (s).

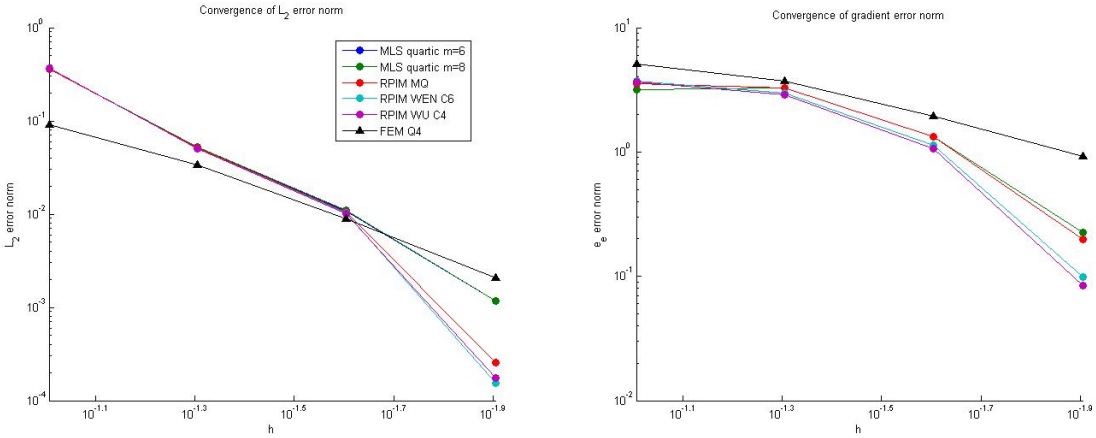


Figure 6.21: Convergence of L_2 error norm(left) and gradient energy norm(right). s=50 and $\alpha_s = 2.0$

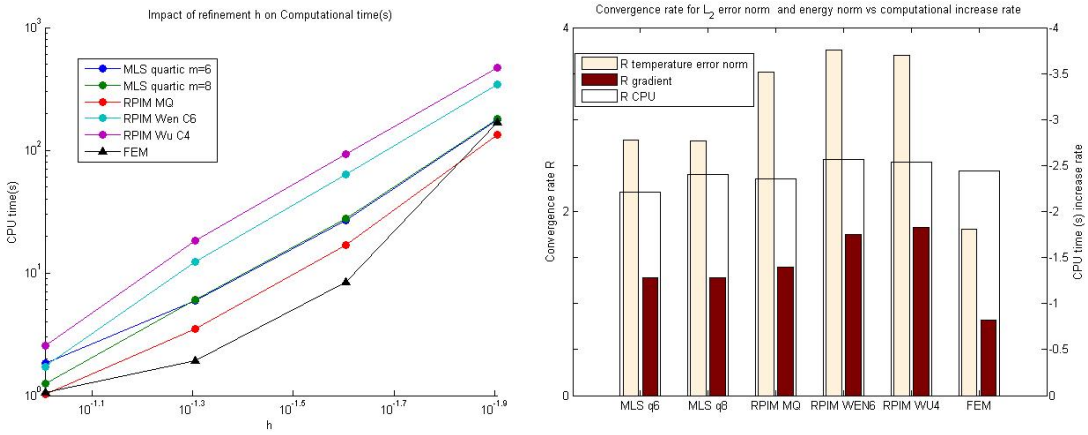


Figure 6.22: Influence of refinement h on Comparison of convergence rate for the L_2 error norm, gradient norm and CPU cost

We also study the impact of refining local nodes around the high gradient area. We use a band of higher density nodes around y=3. The same distribution of 10×60 nodes is used for FEM and EFG. We vary the density h of 30 nodes in a band area around y=3 as depicted in 6.23.

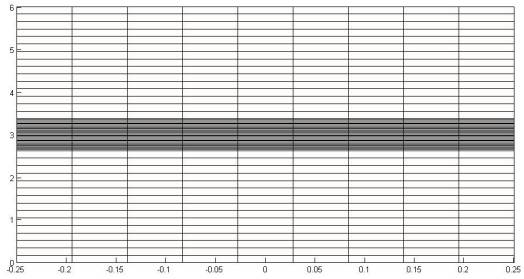


Figure 6.23: Refinement pattern

Figure 6.24, which shows the influence of refining the density h of nodes within the local area. A higher density of nodes in the region of high gradient provides better accuracy for both FEM and EFG.

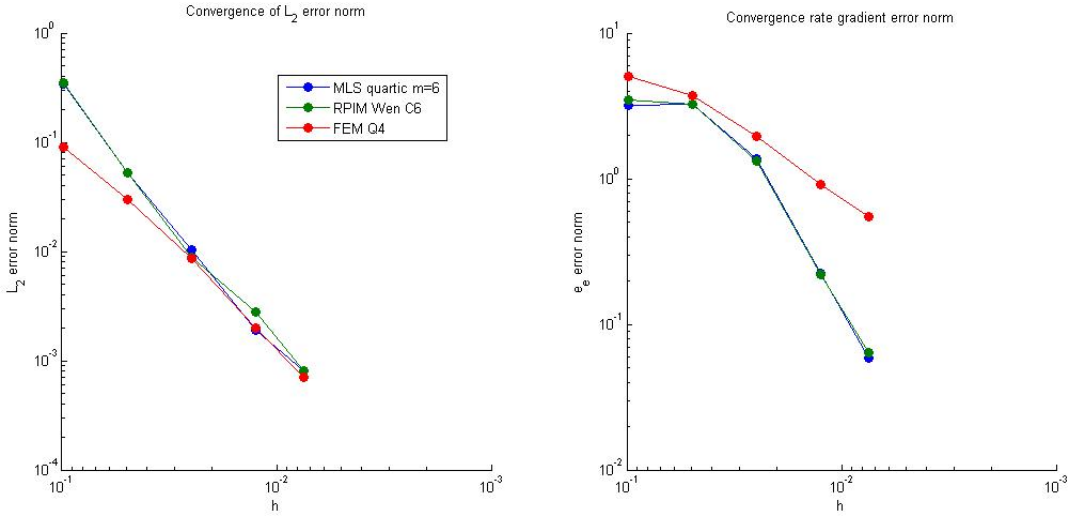


Figure 6.24: Convergence of L_2 error norm and gradient energy norm with local refinement around y=3. s = 50 and $\alpha_s = 2.0$

A convergence rate similar to FEM is obtained for the L_2 error on the temperature field. It is worth noting that EFG still exhibits a higher rate of convergence for the gradient error norm (figure 6.25).

This shows that the refinement of local density nodes works similarly as in FEM. However as the density of nodes become smaller, the selection of nodes for the support domain between areas with a strong density gradient can be tricky. In RPIM, this can lead to poor conditioning of the moment matrix which adversely impacts the accuracy of the results. Another pitfall lies in the underlying grid for quadrature. As the local density varies, the integration cell must adapt in a similar pattern. Failure to do so can dramatically reduce the accuracy and convergence. In the case of complex refinement topology, although the meshfree approximation does not rely on mesh, the need for an adapted background cell for integration introduces similar issues as mesh refinement. This further emphasises the importance of the quadrature issue for EFG method.

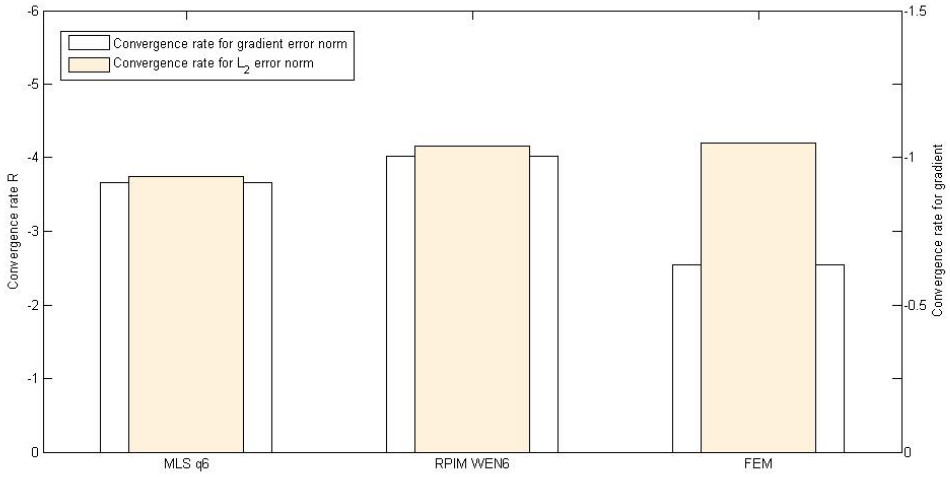


Figure 6.25: Refinement pattern (left) and rate of convergence (right)

6.3 3D heat conduction problems

In respect of the features of meshfree methods highlighted in the previous chapter, it seems that a huge advantage of meshfree methods over mesh based methods is their ability to be extended to tackle 3D problems with less effort. Building a 3D mesh for practical problems is not an easy task. At first sight, the nodal approach seems to be a strong argument for method such as EFG. Imposing essential boundary conditions in 3D becomes a complicated task for meshfree SFs that do not possess the delta property. Either with the penalty method or Lagrange multipliers, the boundary surface integration for complex boundary surfaces in 3D will considerably impact the computational cost of the methods.
Figure 6.26 illustrates the nodal distribution and background cell associated over a 3D slab domain Ω for a heat conduction problem.

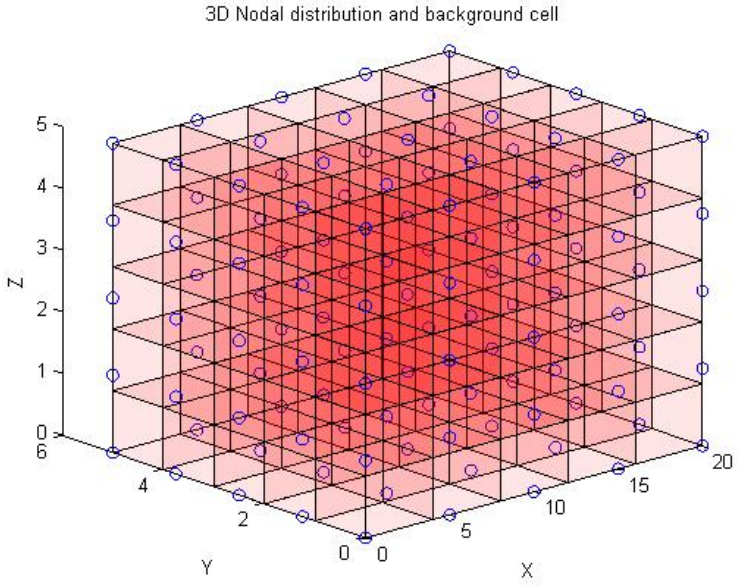


Figure 6.26: Example of $5 \times 5 \times 5$ 3D regular distribution of field nodes and fitted background integration cell

As a first benchmark for 3D problems, we study the following Laplace problem

$$\nabla^2(T) = f \quad \forall(x, y, z) \in \Omega = [0, a] \times [0, b] \times [0, c] \tag{6.28}$$

where f is the source such as

$$f = (\frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2})\pi^2 \sin(\frac{\pi x}{a}) \sin(\frac{\pi y}{b}) \sin(\frac{\pi z}{c}) \tag{6.29}$$

Together with the following essential boundary condition

$$T(0, y, z) = T(a, y, z) = T(x, 0, z) = T(x, b, z) = T(x, y, 0) = T(0, y, z) = 0; \tag{6.30}$$

The exact solution of such problems is given by

$$T(x, y, z) = \sin(\frac{\pi x}{a}) \sin(\frac{\pi y}{b}) \sin(\frac{\pi z}{c}) \tag{6.31}$$

Figure 6.27 shows a comparison of the temperature distribution obtained with the RPIM MQ and the analytical solution of the problem. The Penalty method is used to enforce the Dirichlet bound-

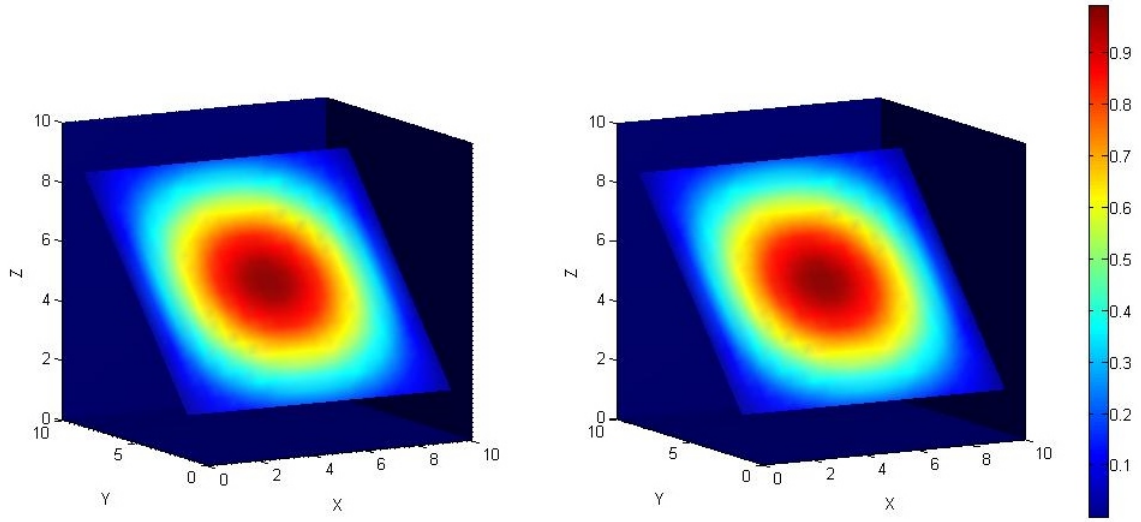


Figure 6.27: Comparison of temperature distribution through a slice view between RPIM MQ(left) and analytical solution(right). $20 \times 20 \times 20$ regularly distributed field nodes. 3x3x3 Gauss points per integration cell.α_s=2. L_2 error =5.04 $\times 10^{-4}$

ary condition. The usual settings for the meshfree shape function are used: $\alpha_c = 3$, $Q = 0.95$ and $\alpha_s = 2$. Figure 6.28 shows the convergence of EFG and RPIM for regular and irregular nodal distributions. We use cubic weight function with m=10 for MLS and MQ RBF for RPIM. Both methods work well with the RPIM being more accurate. This is mainly due to the use of the penalty method to enforce approximately the essential boundary condition in the case of the MLS SF. However for highly irregularly distributed nodes, we found that the convergence is much less as compared to nodes that are regularly or even irregularly distributed with minor perturbations. Figure 6.29 shows the effect of increasing the density of the nodal distribution on the total computational time(blue) and the time required for applying essential boundary condition(red). This hilights the fact that in EFG, the increase in computational cost for 3D problem, doesn't really come from the treatment of the boundary surface condition. The intgration over quad cell in 3D

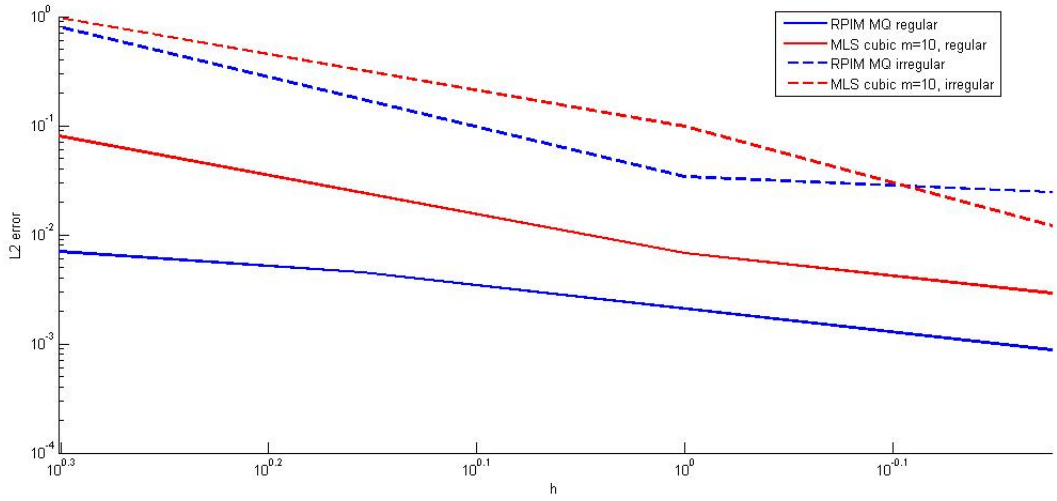


Figure 6.28: Comparison of convergence of L2 error for EFG MLS and RPIM. 3x3x3 Gauss points per integration cell.

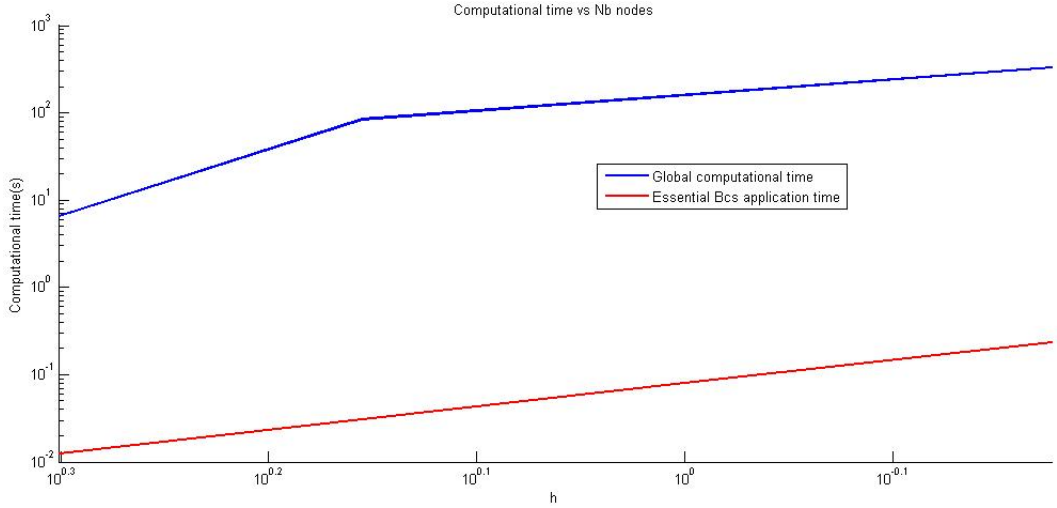


Figure 6.29: Computational time(s) vs h refinement

necessitates to, at least, double the Gauss points. This result in serious increase in computational time.

As a second benchmark for 3D heat conduction problems, we consider the homogeneous Dirichlet Laplace problem on a unit cube which can be exemplified by the following equation

$$\nabla^2(T) = 0 \quad \forall(x, y, z) \in \Omega = [0, 1] \times [0, 1] \times [0, 1] \tag{6.32}$$

Together with the following essential boundary conditions

$$T(0, y, z) = \sin(\pi y) * \sin(\pi z); \tag{6.33}$$

$$T(1, y, z) = 2\sin(\pi y) * \sin(\pi z); \tag{6.34}$$

$$T(x, 0, z) = T(x, 1, z) = T(x, y, 0) = T(x, y, 1) = 0 \tag{6.35}$$

The exact solution of such problem is given by

$$T(x,y,z) = \frac{\sin(\pi y) * \sin(\pi z)}{\sinh(\pi\sqrt{2})} \left(2\sinh(\pi\sqrt{2}x) + \sinh(\pi\sqrt{2}(1-x))\right) \tag{6.36}$$

Figure 6.30 shows the final temperature distribution obtained using $5 \times 5 \times 5$ regularly distributed field nodes. The solution is re-interpolated using MF RPIM MQ interpolant over $25 \times 25 \times 25$ sampling points. The final L_2 error is 0.02. Figure 6.31 shows the convergence for EFG and RPIM

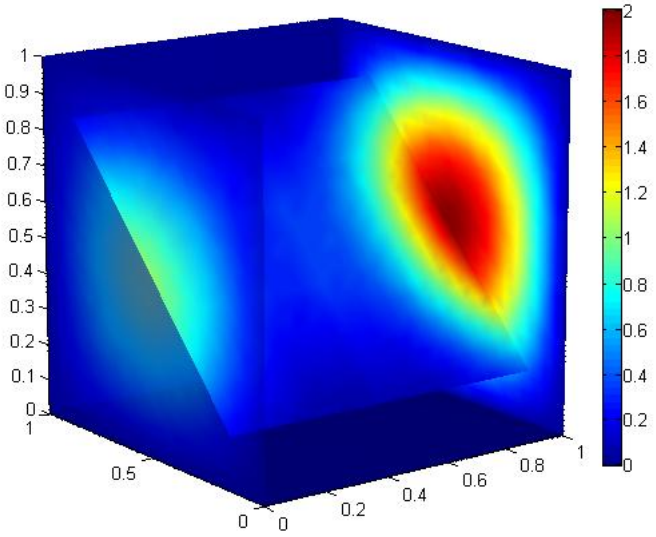


Figure 6.30: Temperature distribution through a slice obtained with RPIM MQ with 5x5x5 regularly distributed field nodes. 3x3x3 Gauss points per integration cell.α_s=2. L_2 error =0.02

for both the regular and irregular distribution of nodes. Figure 6.32 shows the effect of increasing

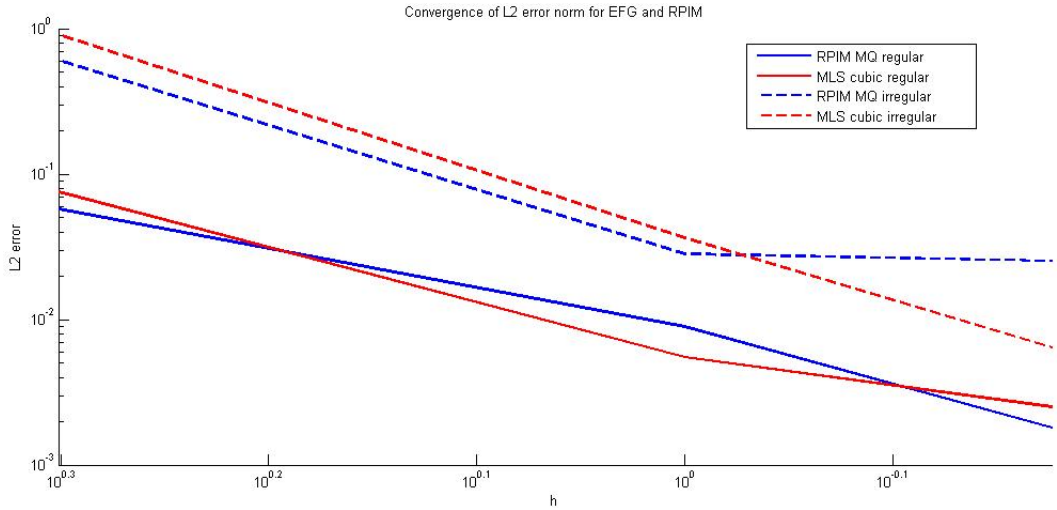


Figure 6.31: Comparison of convergence of L2 error for EFG MLS and RPIM. 3x3x3 Gauss points per integration cell.

the nodal distribution density on the total computational time and the time required for applying essential boundary conditions.

Finally, we consider the Laplacian problem with the following Neumann boundary condition.

$$u_x(0,y,z) = u_x(1,y,z) = u_y(x,0,z) = u_y(x,1,z) = 0 \tag{6.37}$$

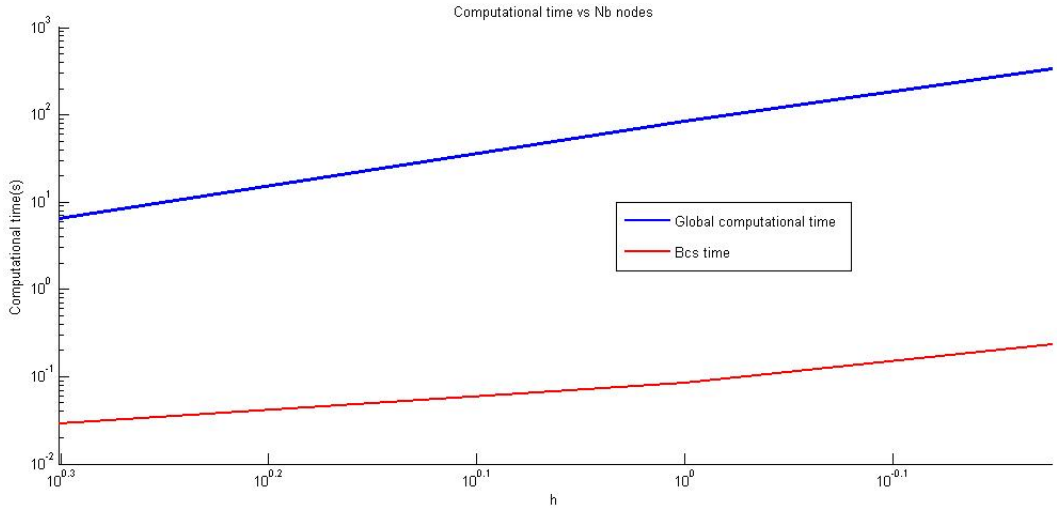


Figure 6.32: Computational time(s) vs h refinement

and

$$u_z(x, y, 0) = \cos(\pi x)\cos(\pi y); \quad and u_z(x, y, 1) = 0 \tag{6.38}$$

Figure 6.33 shows a comparison of the temperature distribution obtained with the RPIM MQ with a $10 \times 10 \times 10$ regular distribution of nodes, and the analytical solution. The solution is re-interpolated at $25 \times 25 \times 25$ using the same RPIM interpolant. Similar results are obtained for this

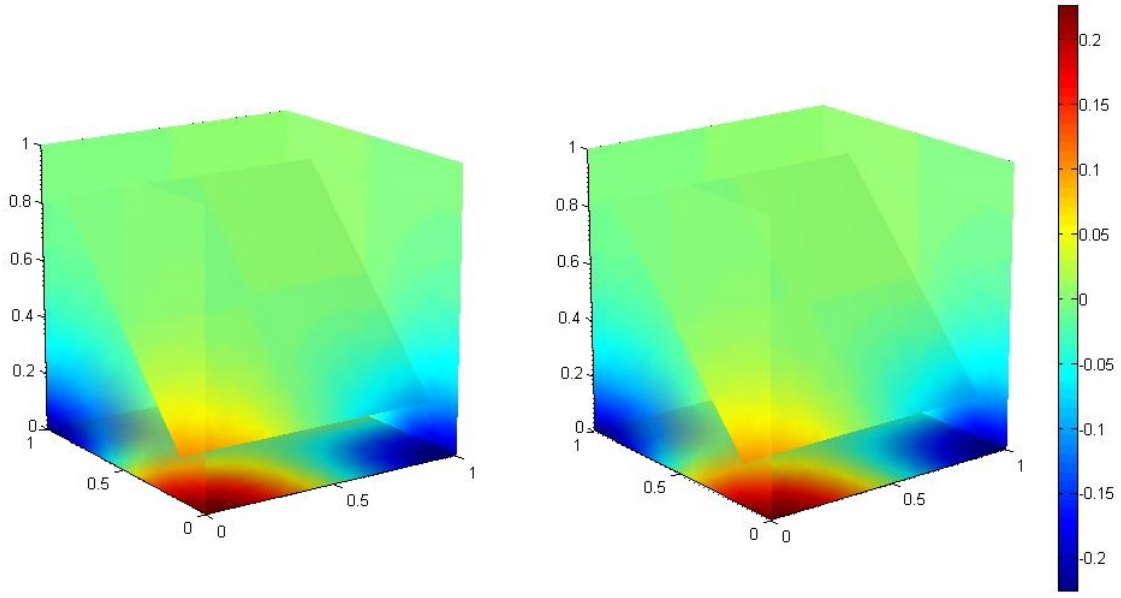


Figure 6.33: Comparison of the temperature distribution with Neumann type boundary condition, through a slice view between RPIM MQ (left) and analytical solution (right). $10\times10\times10$ regularly distributed field nodes. 3x3x3 Gauss points per integration cell.α_s=2. L_2 error =8,7$\times10^{-3}$

problem.The error obtained for a $10 \times 10 \times 10$ is 8.7×10^{-3}.

6.4 Transient homogeneous heat conduction problem

In MEMS design, transient thermal problems are of a great interest as many thermal actuators are mechanical-thermal based. Heat conduction is also of prime importance for many packaging process characterization. The transient phenomenon that is observed during the analysis of diffusion

problems will ultimately reach a steady state beyond a certain point in time. In some cases this steady state is modelled by the Laplace or Poisson equation.
The governing equation for a transient heat conduction problem in a homogeneous and isotropic solid body region Ω is

$$\rho C\frac{\partial T}{\partial t} - k\nabla^2 T - Q(x,t) = 0 \quad \forall x \in \Omega \tag{6.39}$$

T is the temperature scalar field, k the thermal conductivity of the material, ρ the material density and c the heat capacity ($Jkg^{-1}k^{-1}$). $Q(x,t)$ is the source term which corresponds to the rate of heat flow(W) generated per unit volume and per time unit at the location x. Again we define the boundary conditions as follows:

$$T = \overline{T}, \quad \forall x, y \in \Gamma_1 \tag{6.40}$$

$$\frac{\partial T}{\partial n} = \overline{q} \cdot n, \quad \forall x, y \in \Gamma_2 \tag{6.41}$$

$$k\nabla T \cdot n = h(T_s - T_a), \quad \forall x, y \in \Gamma_3 \tag{6.42}$$

With $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$. As we deal with the initial boundary value problem (IBVP) we also impose the initial condition $T|_{t=0} = T_0$. $T_s = T(\Gamma, t)$ is the temperature at the surface of the solid and T_∞ denotes the temperature of the surrounding environment. The weak formulation can be obtained using weight residual or minimization of energy principle.

$$\int_\Omega \nabla w \cdot (k\nabla T)d\Omega + \int_\Omega \left[w\left(\rho c\frac{\partial T}{\partial t} - Q\right)\right]d\Omega - \int_{\Gamma_2} w \cdot \bar{q}d\Gamma - \int_{\Gamma_3} wh(T_s - T_a)d\Gamma = 0 \tag{6.43}$$

Enforcing the essential boundary condition using the penalty method gives the final weak form

$$\begin{aligned}\int_\Omega \nabla w \cdot (k\nabla T)d\Omega + \int_\Omega \left[w\left(\rho c\frac{\partial T}{\partial t} - Q\right)\right]d\Omega - \int_{\Gamma_2} w \cdot \bar{q}d\Gamma \\ - \int_{\Gamma_3} wh(T_s - T_a)d\Gamma + \int_{\Gamma_1} w\alpha(T - \bar{T})d\Gamma = 0\end{aligned} \tag{6.44}$$

Discretisation needs to be performed for both time and space variables. In the case of space variables only, we first obtain the so-called semi-discrete approximation of the weak formulation equation 6.44. In this semi discrete form, the solution is expressed as

$$T^h(x,t) = \sum_{j=1}^{N} \Phi_j(x) \cdot T_j(t) = \boldsymbol{\Phi(x)} \cdot \boldsymbol{T} \tag{6.45}$$

and

$$\frac{\partial T^h(x,t)}{\partial t} = \frac{\partial}{\partial t}\sum_{j=1}^{N} \Phi_j(x) \cdot T_j(t) = \sum_{j=1}^{N} \Phi_j(x) \cdot \frac{\partial T_j(t)}{\partial t} = \boldsymbol{\Phi(x)} \cdot \dot{\boldsymbol{T}} \tag{6.46}$$

Where

$$\boldsymbol{\phi(x)} = \{\phi_1(x), \phi_2(x), \ldots, \phi_n(x)\} \tag{6.47}$$

$$\boldsymbol{T} = \{T_1(t), T_2(t), \ldots, T_n(t)\}^T \tag{6.48}$$

and

$$\dot{\boldsymbol{T}} = \{\frac{\partial T_1(t)}{\partial t}, \frac{\partial T_2(t)}{\partial t}, \ldots, \frac{\partial T_n(t)}{\partial t}\}^T \tag{6.49}$$

Where Φ_j are the regular basis functions in the space V_h. A separation of variable is applied.

The shape function is used to interpolate the spatial variation while the time variation is related to the nodal variables. Substituting the solution into the semi-discrete weak formulation leads to the following system for T^h.

$$\int_\Omega \nabla\Phi_I \cdot (k\nabla\Phi_J T_J)d\Omega + \int_\Omega \left[\Phi_I \left(\rho c \Phi_J \frac{\partial T_J}{\partial t} - Q\right)\right] d\Omega - \int_{\Gamma_2} \Phi_I \cdot \bar{q} d\Gamma \\ - \int_{\Gamma_3} \Phi_I h \Phi_J (T_J - T_a) d\Gamma + \int_{\Gamma_1} \Phi_I \alpha \Phi_J (T_J - \bar{T}) d\Gamma = 0 \tag{6.50}$$

equivalent to the final system

$$\boldsymbol{C\dot{T} + KT = F} \tag{6.51}$$

With

$$C_{IJ} = \int_\Omega \Phi_I \rho c \Phi_J d\Omega \tag{6.52}$$

$$K = \int_\Omega \nabla\Phi_I \cdot (k\nabla\Phi_J)d\Omega - \int_{\Gamma_3} \Phi_I h \Phi_J d\Gamma + \int_{\Gamma_1} \Phi_I \alpha \Phi_J d\Gamma \tag{6.53}$$

$$F = \int_\Omega \Phi_I Q d\Omega + \int_{\Gamma_2} \Phi_I \cdot \bar{q} d\Gamma + \int_{\Gamma_3} \Phi_I h T_a d\Gamma + \int_{\Gamma_1} \Phi_I \alpha \bar{T} d\Gamma \tag{6.54}$$

C is called the capacitance matrix. It contains the product of the density ρ and the specific heat. At this stage only spatial discretisation has been achieved and the discretisation regarding time can be done by partitioning the time interval [0,T] into N equal intervals $[t_k, t_{k+1}]$ of length $\Delta t = \frac{T}{N}$. There are many established time-marching numerical methods available to deal with time-dependent behaviour. Amongst them, the most popular are the explicit and implicit methods. The general approach it to consider the slope at the initial point T_t and the slope at an intermediate point $T_{t+\theta\Delta t}$. They are used to obtain the function at the end of the time step $T_{t+\theta\Delta t}$. The general 2 points FD scheme is expressed

$$\theta \left(\frac{\partial T}{\partial t}\right)_{t+\Delta t} + (1-\theta)\left(\frac{\partial T}{\partial t}\right)_t = \frac{T_{t+\Delta t} - T_t}{\Delta t} \tag{6.55}$$

Using equation 6.51 for instant t and $t + \Delta t$, we have

$$(1-\theta)C\dot{T}_t = (1-\theta)(F_t - K_t T_t) \tag{6.56}$$

and

$$\theta C\dot{T}_{t+1} = \theta(F_{t+1} - K_{t+1}T_{t+1}) \tag{6.57}$$

Substituting 6.56 and 6.57 into 6.55 gives the following general form of discrete equation

$$\left(\frac{C}{\Delta t} + \theta K_{t+1}\right) T_{t+1} = \left(\frac{C}{\Delta t} - (1-\theta)K_t\right) T_t + \theta F_{t+1} + (1-\theta)F_t \tag{6.58}$$

Depending on the values of θ, the formulation leads to different form. When $\theta = 0$ we obtain the explicit forward difference scheme or Euler method

$$\left(\frac{C(T_{t+1} - T_t)}{\Delta t}\right) + K_t T_t = F_t \tag{6.59}$$

This scheme is known as "Forward-difference" because it arrives at the new value of the temper-

ature, $T_{t+\Delta t}$ by moving forward along the tangent at the previous point T_t. The new temperature values are calculated from the old ones and there is no $T_{t+\Delta t}$ in the right hand side of the equation 6.59;thus the term explicit. When $\theta = 0.5$, it leads to the Crank-Nicholson formulation

$$\left(\frac{C(T_{t+1} - T_t)}{\Delta t} + \frac{1}{2}K_{t+1}\right) T_{t+1} + K_t T_t = \frac{1}{2}(F_{t+1} + F_t) \tag{6.60}$$

If $\theta = 2/3$, the Galerkin form is obtained

$$\left(\frac{C(T_{t+1} - T_t)}{\Delta t} + \frac{2}{3}\theta K_{t+1}\right) T_{t+1} + \frac{1}{3}K_t T_t = \frac{2}{3}F_{t+1} + \frac{1}{3}F_t \tag{6.61}$$

When $\theta > 0.5$, the scheme is stable for all Δt. Finally for $\theta = 1$, the slope at the future temperature $T_{t+\Delta}$ is used. It leads to the implicit method

$$\left(\frac{C(T_{t+1} - T_t)}{\Delta t} + \theta K_{t+1}\right) T_{t+1} = F_{t+1} \tag{6.62}$$

The implicit method is unconditionally stable. This equation has to be solved for $k = 0, 1, \cdots, N-1$ with u_k^h being the approximation of $u(t)$ at the time step $t_k = k\Delta t$. This is a similar formulation that FEM uses [63]. The value of θ has important implications for the stability of the scheme.

$$\left(\frac{C}{\Delta t} + \theta K_{t+1}\right) T_{t+1} = \left(\frac{C}{\Delta t} - (1-\theta)K_t\right) T_t + \theta F_{t+1} + (1-\theta)F_t \tag{6.63}$$

Using the Crank Nicholson scheme for instance gives the complete discretised system necessary to compute $T_{t+\Delta t}^h$

$$(\frac{1}{2}\Delta t K_{t+\Delta t} + C)T_{t+1} = \frac{1}{2}\Delta t(F_{t+1} + F_t) + (C - \frac{1}{2}\Delta t K_t)T_t \tag{6.64}$$

In the case of a linear problem the conductance matrix does not vary with time.
We consider the following transient heat conduction problem in the 2D domain $\Omega = [0, \pi] \times [0, \pi]$ governed by the following IBVP

$$u_t = u_{xx} + u_{yy}, \quad \forall x \in \Omega, t > 0 \tag{6.65}$$

$$u_{,x}(0, y, t) = u_{,x}(\pi, y, t) = 0; \tag{6.66}$$

$$u(x, 0, t) = 0, \quad u(x, \pi, t) = 0; \tag{6.67}$$

$$u(x, y, 0) = \cos(x)\sin(y) + \cos(2x)\sin(2y), \tag{6.68}$$

The analytical solution for this problem is given by

$$u(x, y, t) = e^{-2t}\cos(x)\sin(y) + e^{-8t}\cos(2x)\sin(2y); \tag{6.69}$$

Figure 6.34 show the temperature distribution at t=0.1 s obtained with MLS shape function with cubic weight function, for ds=1.5 and 30×30 regular node distribution. The $L_2 = 8.7 \times 10^{-4}$. Using both C-N and Galerkin scheme show that the method converges quite quickly as the time step decreases. Figure 6.35 shows the result for MLS(bleu) and RPIM-MQ(green). From our test, a time step around 0.01 s gives the best results.

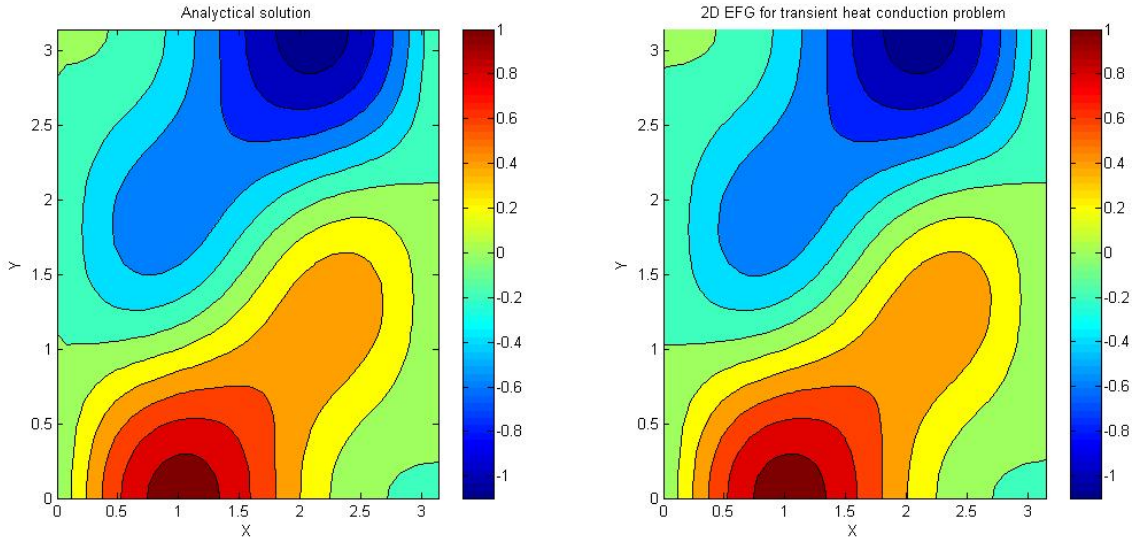


Figure 6.34: Comparison of analytical solution (left) with EFG solution (right) at t=0.1s for Δt=0.001s. L_2 error is 0.08%

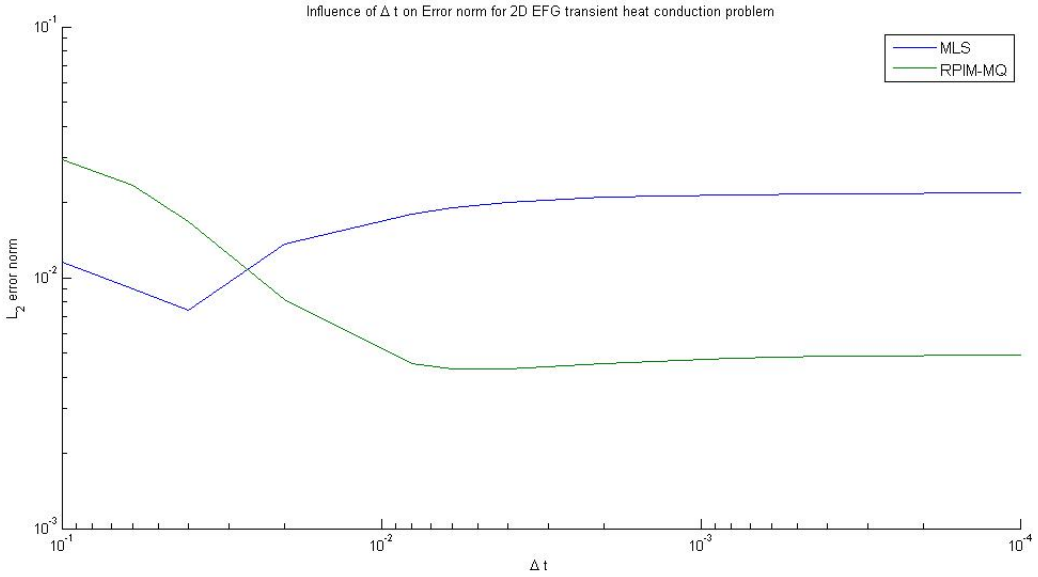


Figure 6.35: Influence of time step Δt on convergence of solution with EFG solution

Figure 6.36 shows the convergence for 0.1s for the EFG with MLS. Similar convergence is obtained with RPIM method. Overall, meshfree based on weak formulation works well for transient heat conduction problem. No increase in computational cost over regular mesh based methods have been found through our tests. This is mainly due to the fact that the separation of variable allow to treat the time derivative in equation 6.39 in similar manner as it is done in FEM.

6.5 Time dependent heat conduction with temperature dependent thermal properties

MEMS design deals with a great number of temperature dependent problems. Heat conduction phenomena in which heat capacities and thermal conductivities depend on temperature are part of the many design process of MEMS. From the characterisation of sensors, to the use of thermal material properties for MEMS design, modelling heat conduction for complex associations of

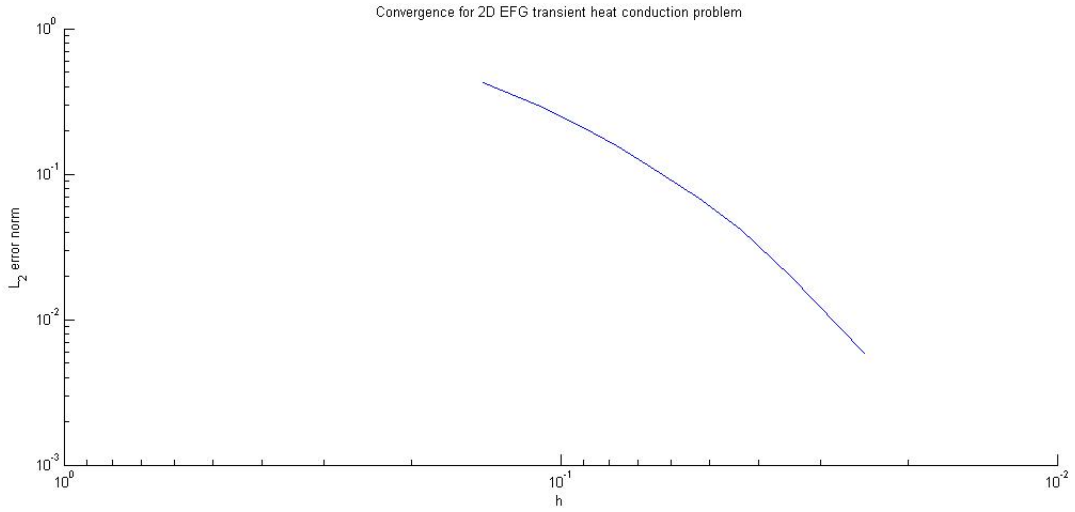


Figure 6.36: Convergence of L_2 error norm at t=0.1s for $\Delta t = 0.01s$.

material with non-linear thermal properties is of great interest for MEMS designers. The study of thermal effects is important in MEMS design as fluctuations in ambient and operating temperature have a drastic impact on strain measuring circuit [1]. Thermal properties for material use in MEMS fabrication or smart materials are not linear function of temperature. We demonstrate an approach for non-linear thermal properties that can be used for Meshfree weak based methods.
We consider the 2D Heat conduction problems with temperature dependent heat capacities and thermal conductivities over a domain Ω. IT is described by the following equation

$$\rho C(T)\frac{\partial T}{\partial t} - \nabla(k(T)\nabla T) - Q(x,t) = 0 \quad \forall x \in \Omega \tag{6.70}$$

$$T(x,t) = \overline{T}(x,t), \quad \forall x \in \Gamma_1 \tag{6.71}$$

$$k(T)\frac{\partial T}{\partial n} = \overline{q}(x,t)\cdot n, \quad \forall x \in \Gamma_2 \tag{6.72}$$

$$k\nabla T \cdot n = h(T_s - T_a), \quad \forall x \in \Gamma_3 \tag{6.73}$$

and

$$T(x,0) = \overline{T_0}(x), \quad \forall x \in \Omega \tag{6.74}$$

Using Kirchhoff transformation we can write

$$u(x,t) = \int_{T_r}^{T} k(T)dT \tag{6.75}$$

we can rewrite 6.70,6.71,6.72 and 6.74 into

$$\gamma(u)\frac{\partial u}{\partial t} - \nabla^2 u - Q(x,t) = 0 \quad \forall x \in \Omega \tag{6.76}$$

with$\gamma = \rho C/k$ called the reciprocal of thermal diffusivity.

$$u(x,t) = \overline{u}(x,t), \quad \forall x \in \Gamma_1 \tag{6.77}$$

$$\frac{\partial u}{\partial n} = \overline{q}(x,t)\cdot n, \quad \forall x \in \Gamma_2 \tag{6.78}$$

$$\nabla u \cdot n = h(T_s - T_a), \quad \forall x \in \Gamma_3 \tag{6.79}$$

and

$$u(x,0) = \overline{u}_0(x), \quad \forall x \in \Omega \tag{6.80}$$

We consider the thermal properties of several materials such as silicon (Si), zirconium(Zr), tantalum (Ta) and tungsten(W) in the temperature range 100 to 1200 K. For simplicity, we assume that constant density is given at $20°C$ being 6570, 19300 and 16600 $kg.m^{-3}$ respectively. But a similar strategy can be adopted in dealing with the variation of this property with temperature. Figure 6.37 gives the thermal properties, heat Capacity in $J.Kg^{-1}.K^{-1}$ and the thermal conductivity in $W.m^{-1}.K^{-1}$ for the materials at different temperatures samples.

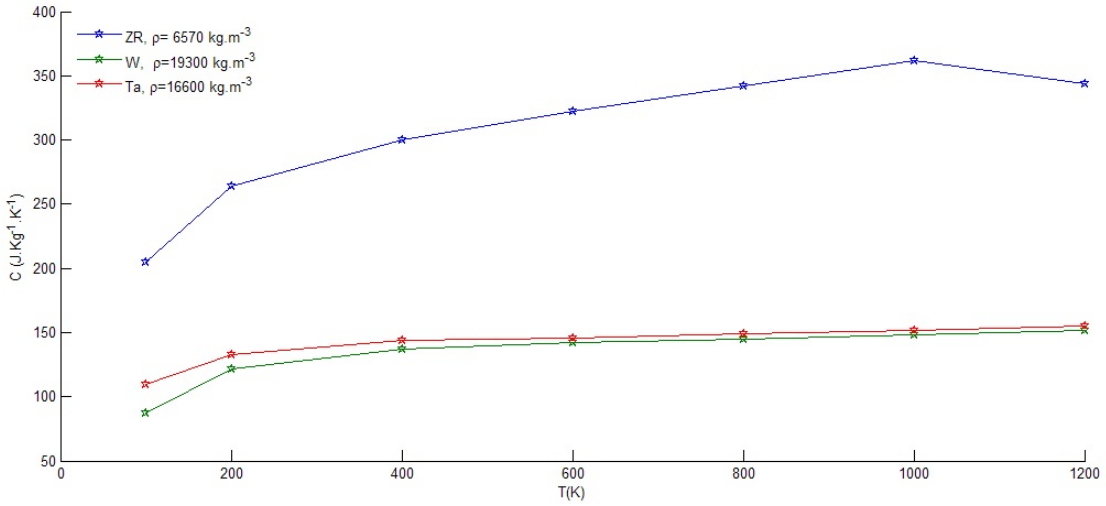


Figure 6.37: Specific heat capacity C for zirconium, tungsten and tantalum at different temperature sampling values

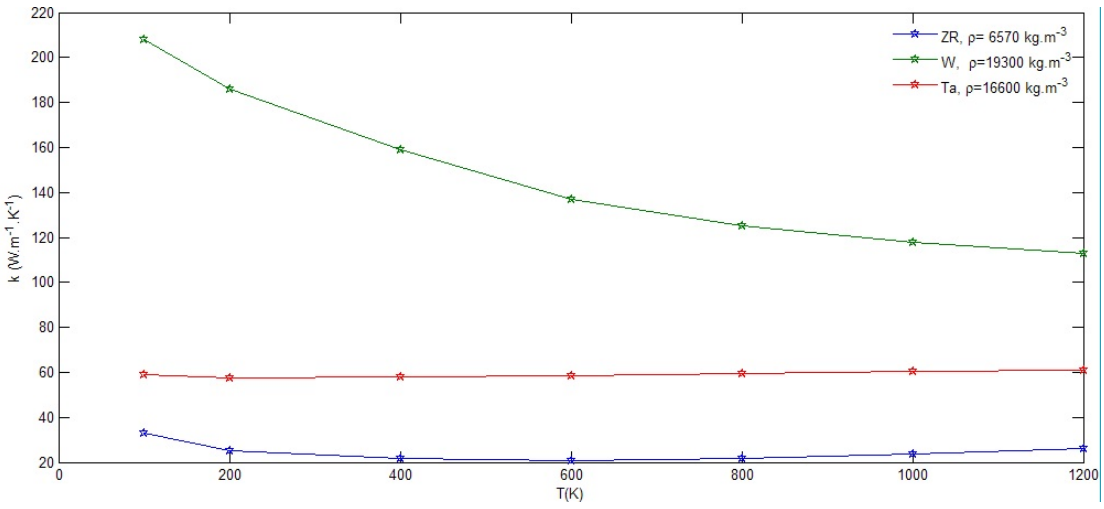


Figure 6.38: Thermal conductivity k for zirconium, tungsten and tantalum at different temperature sampling values

This set of data can be obtained from any material suppliers such as Plantsee [221]. From equation 6.75, u(T) appears as a monotonically increasing function due to the fact that k(T) is always positive. Finding an explicit form for u(T) is always possible on the opposite of T(u) unless k(T)

is of linear form. We approximate k(T) using a piecewise linear function

$$k(T) = k_{i+1} + \frac{k_i - k_{i-1}}{T_i - T_{i-1}}(T - T_{i-1}), \quad \forall T \in [T_{i-1}, T_i[\tag{6.81}$$

$T_i, i = 1...N$ are temperature sampling values from the material model (see figure6.37 and see figure6.38). We take the temperature reference $T_r = 100k$ and substitute k(T) from equation 6.81 into 6.75, it gives the following recurrence form for u(T)

$$u(T) = u_{i-1} + k_{i-1}(T - T_{i-1} + \frac{1}{2}\left(\frac{k_i - k_{i-1}}{T_i - T_{i-1}}\right)(T - T_{i-1})^2 \tag{6.82}$$

We use 7 samples values for T_i and compute the Kirchhoff variables $u(T)$ as shown on Figure 6.39.

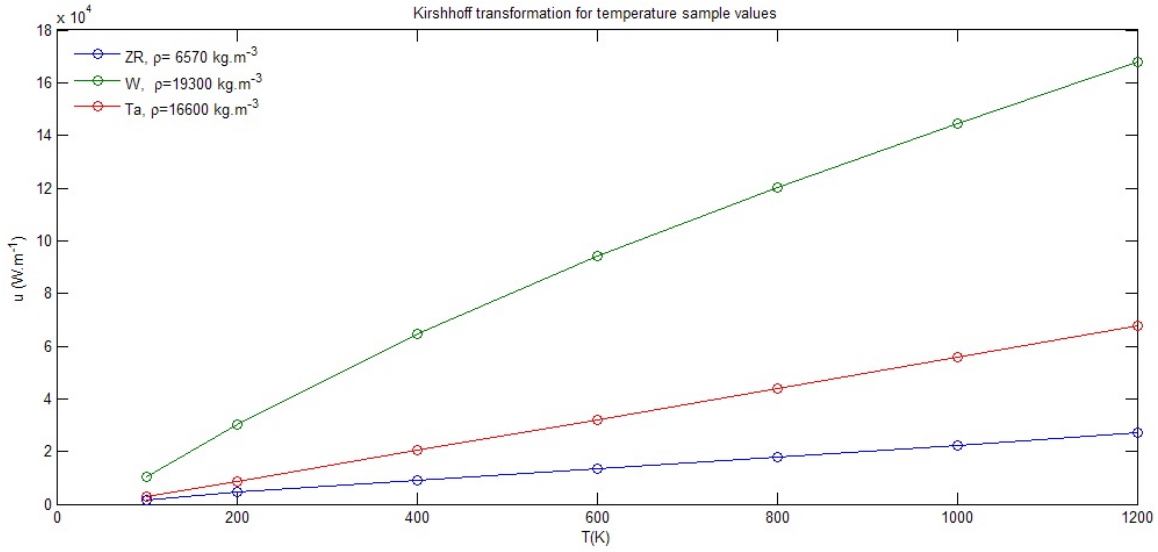


Figure 6.39: Kirchoff variable $u(T)$ values for Zr, W and Ta at different temperature sampling values.

Having computed the values of Kirchhoff variable for each temperature samples, we can calculate the reciprocal of thermal diffusivity γ which corresponds to each $u(T_i)$. Figure 6.40 shows the computed reciprocal values of thermal diffusivity for the different material considered.
In order to obtain $\gamma(u)$ at each iteration, a piecewise linear approximation is used. From equation 6.81 the inversion formula for T(u) can be derived. For $u_{i-1} \leq u < u_i$ we have

$$T(u) = T_{i-1} + \left(\frac{T_i - T_{i-1}}{k_i - k_{i-1}}\right)\left[-k_{i-1} + \sqrt{k_{i-1}^2 + 2\left(\frac{k_i - k_{i-1}}{T_i - T_{i-1}}\right)(u - u_{i-1})^2}\right] \tag{6.83}$$

At each iteration, the resulting temperatures u^n are compared to the prior estimate of the nodal temperatures. If the difference between the prior and resulting temperatures is acceptable, as defined below, then the analysis iterations can conclude. If the difference is unacceptable, then the thermal conductivity and radiation heat rates are recalculated based on the newly assumed nodal temperature estimates and the problem is solved again. The iteration procedure is continued until the temperature increment at nodes is sufficiently small such as $\Delta T^n \leq \epsilon_a$. We use the average

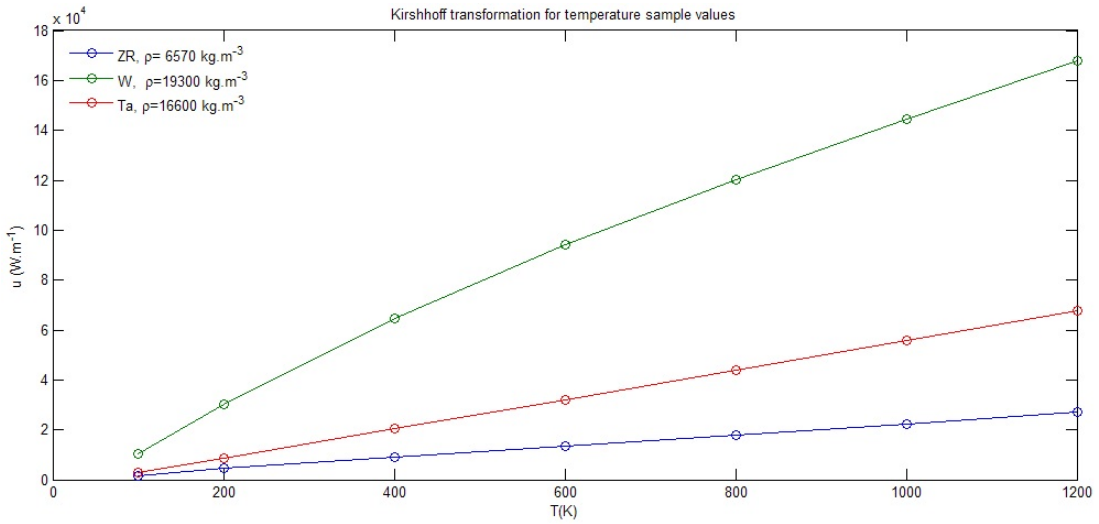


Figure 6.40: Reciprocal of thermal diffusivity γ for Zr, W and Ta at different temperature sampling values.

differences between new and old values of u^n.

$$\sqrt{\frac{1}{N_i}\sum_{i=1}^{Nin}\left(1-\frac{{}^{new}u_i^n}{{}^{old}u_i^n}\right)^2} < \epsilon \tag{6.84}$$

The tolerance criterion ϵ must be carefully set to ensure convergence. In the following, ϵ is taking as $1.0^{\times}10^{-5} \leq \epsilon \leq 1.0^{\times}10^{-4}$

The problem domain Ω considered is a square plate of 0.12×0.12 m, top, bottom and right surfaces are subject to a prescribed temperature. Left surface is under a prescribed flux. The initial boundary conditions are given by the exact solution of such a problem

$$T_{exa}(x,y,t) = (104.43 + 4.43\cos(0.01t))\, e^{(10(x+y))} \tag{6.85}$$

The source function is taken as

$$Q(x,y,t) = \rho C(T)\frac{\partial T}{\partial t} - k(T)\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}\right) - \frac{\partial k}{\partial T}\left[\left(\frac{\partial^2 T}{\partial x^2}\right)^2+\left(\frac{\partial^2 T}{\partial y^2}\right)^2\right] \tag{6.86}$$

Figure 6.41 shows a comparison of the temperature profile obtained at t=300s with meshfree method and the analytical solution using 5×5 regularly distributed nodes. result are pretty good and congruent to those obtain for linear problems. Figure 6.42 shows the results obtained for a 5×5 nodes distribution for L2 error norm(red curve on the left) and the gradient of the error norm(right). MLS cubic shape function is used for this test. The red blue and black curve are respectively the L_2 error for Δt = 0.001, 0.1 and 0.5.

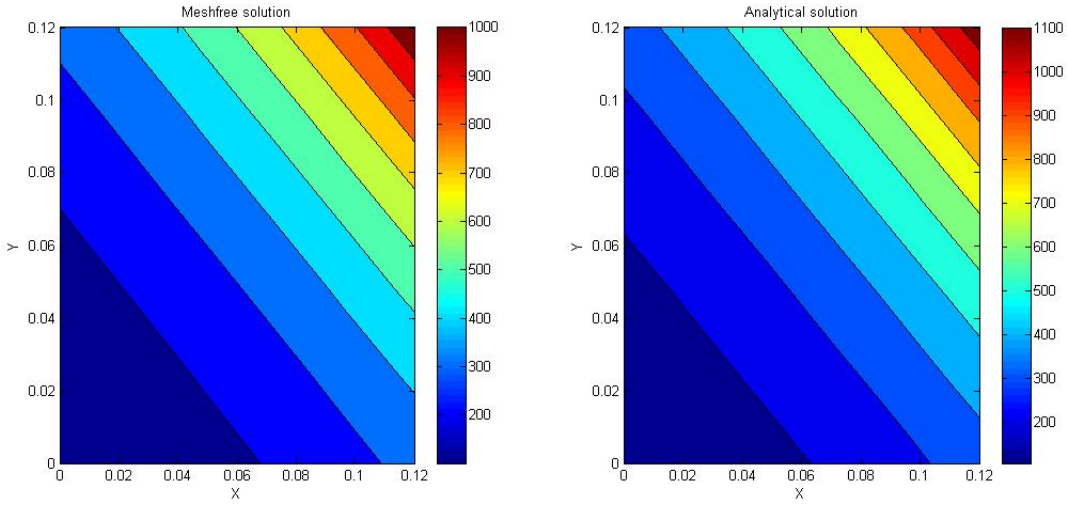


Figure 6.41: Comparison of the temperature at t=300 s, obtained by meshfree method (left) and the analytical solution (right)

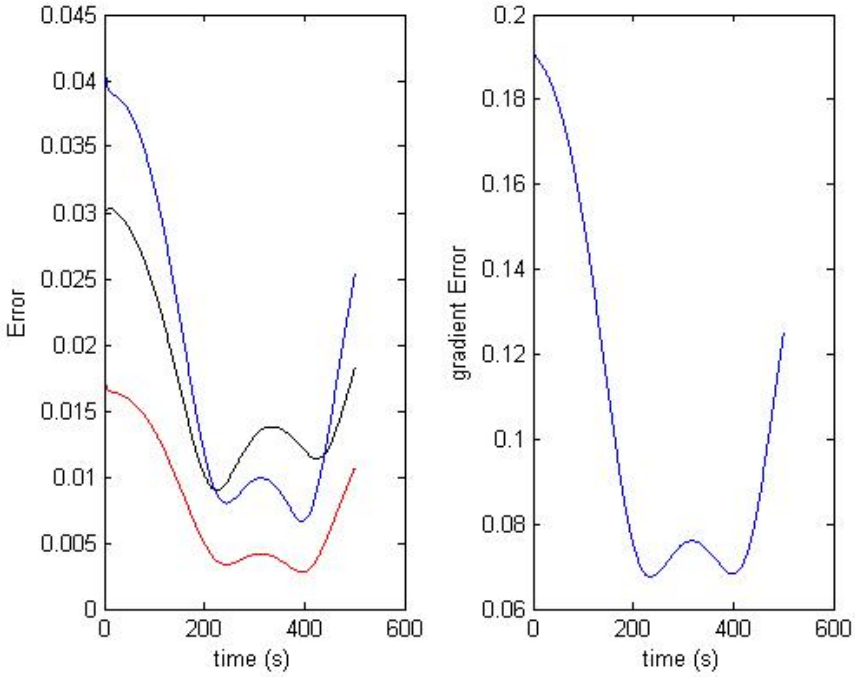


Figure 6.42: Convergence of L_2 error norm

Figure 6.43 represents the evolution of teh reciprocal diffusivity γ, the temperature (middle) and its gradient(right) at the center of the plate. Red curve represent the analytical solution and blue cruve the approximated solution obtained by EFG using MLS with cubic spline weight function. A similar result with a 10×10 nodes distribution is shown on figure 6.44. Different Error norm are used.

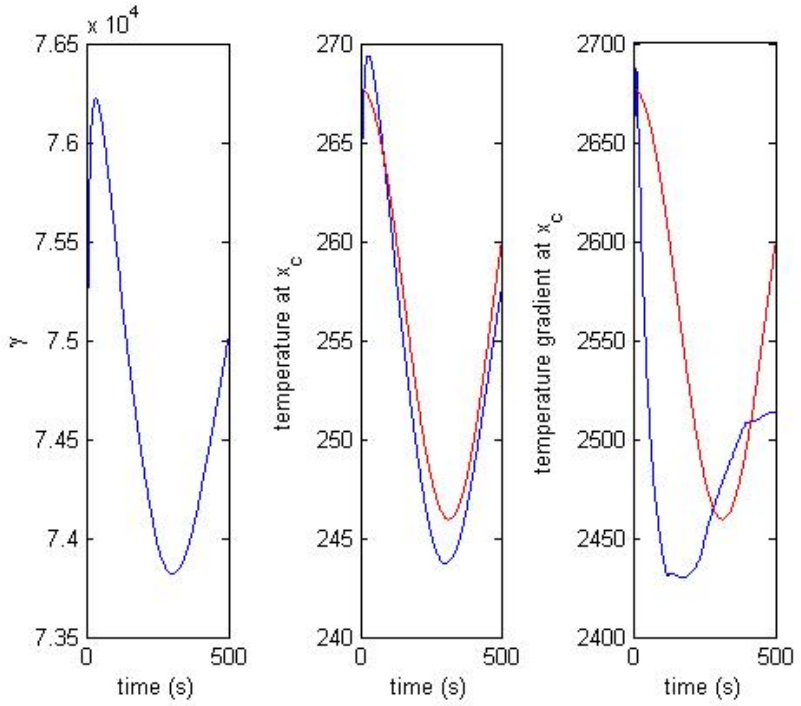


Figure 6.43: Variation of thermal diffusivity γ (left), temperature at center point of the plate (center) and the temperature gradient (right)

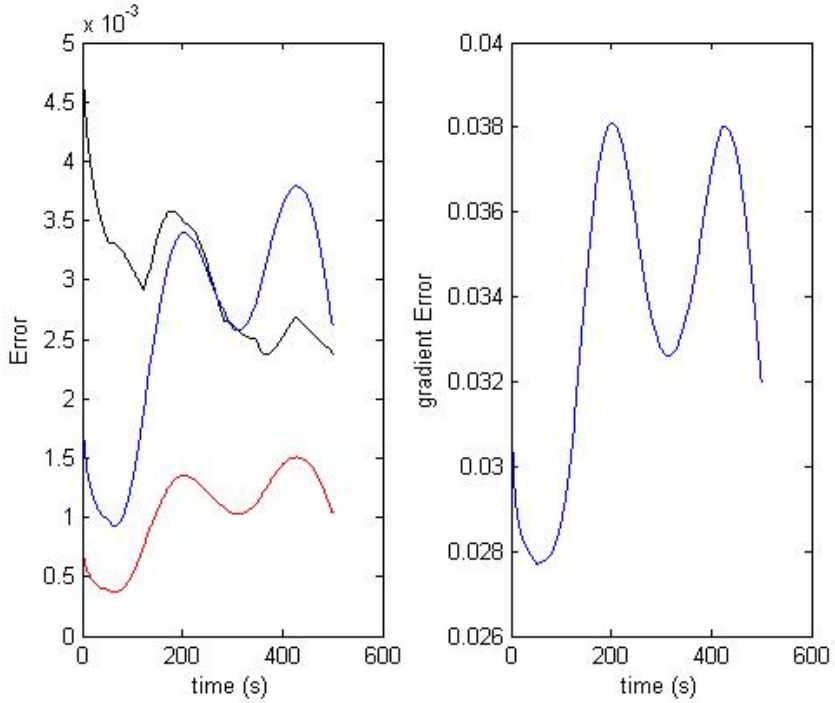


Figure 6.44: L2 error norm (Black), L1 error norm (blue),Relative norm (red).

This show the applicability of global weak based methods for Non-linear temperature problems which are recurrent in MEMS design. However due to the computational cost, it appears difficult to integrate the present method for real time haptic feedback without further improvement or simplification.

6.6 2D elasto-dynamic problems

Elasto-dynamics dealing with solids and structures experiencing vibration under the action of dynamic forces which vary with time are also an important study area for MEMS actuators design and MEMS packaging. Using equilibrium equation between stress and external forces, the governing equation for dynamic equilibrium gives the following linear 2D strong form in tensor notation, for elasto-dynamic problems

$$\sigma_{ii,j} + b_i = m\ddot{u}_i + c\dot{u}_i, \tag{6.87}$$

Where m is the mass density and c the damping coefficient. Obviously $\ddot{u}_i = \frac{\partial^2 u_i}{\partial t^2}$ is the acceleration and $\dot{u}_i = \frac{\partial u_i}{\partial t}$ is the velocity. The boundary condition is given by

$$u_i = \bar{u}_i \quad on \ \Gamma_u \tag{6.88}$$

$$\sigma_{ij} n_j = \bar{t}_i \quad on \ \Gamma_t \tag{6.89}$$

and the initial Bcs for the displacement and the velocity are given by

$$u(x, t_0) = u_0(x), \quad \forall x \in \Omega \tag{6.90}$$

$$\dot{u}(x, t_0) = v_0(x), \quad \forall x \in \Omega \tag{6.91}$$

Using either the Lagrangian principle of least action(see sectionB.3) or minimising principle along with the penalty method to enforce the boundary condition, the following constrained Galerkin form is derived

$$\begin{aligned} &\int_\Omega \delta u^T m\ddot{u} d\Omega + \int_\Omega \delta u^T c\dot{u} d\Omega + \int_\Omega (L\delta u)^T (DLu) d\Omega \\ &- \int_\Omega \delta u^T b d\Omega - \int_{\Gamma_t} \delta u^T \bar{t} d\Gamma + \int_{\Gamma_u} \delta u^T \alpha \left(u - \bar{u}\right) \bar{t} d\Gamma = 0 \end{aligned} \tag{6.92}$$

In order to discretise the above equation, because the trial function for the displacement is a function of space and time, we make a separation of variables. For any time t the displacement u^h and its time derivatives are function of spatial coordinates. Thus using the meshfree shape functions we can write

$$u^h(x, t) = \sum_{j=1}^{n} \Phi_j(x) u_j(t) = \boldsymbol{\Phi(x)} \cdot \boldsymbol{u} \tag{6.93}$$

$$\frac{\partial u^h(x, t)}{\partial t} = \frac{\partial}{\partial t} \sum_{j=1}^{n} \Phi_j(x) u_j(t) = \sum_{j=1}^{n} \Phi_j(x) \frac{\partial u_j(t)}{\partial t} = \boldsymbol{\Phi(x)} \cdot \dot{\boldsymbol{u}} \tag{6.94}$$

and

$$\frac{\partial^2 u^h(x, t)}{\partial t^2} = \frac{\partial^2}{\partial t^2} \sum_{j=1}^{n} \Phi_j(x) u_j(t) = \sum_{j=1}^{n} \Phi_j(x) \frac{\partial^2 u_j(t)}{\partial t^2} = \boldsymbol{\Phi(x)} \cdot \ddot{\boldsymbol{u}} \tag{6.95}$$

With

$$\boldsymbol{\phi(x)} = \{\phi_1(x), \phi_2(x), \ldots, \phi_n(x)\} \tag{6.96}$$

$$\boldsymbol{u} = \{u_1(t), u_2(t), \ldots, u_n(t)\}^T \tag{6.97}$$

Substituting equation 6.93, 6.94 and 6.95 into 6.92 gives

$$\int_\Omega \delta \left(\sum_{I=1}^{n} \Phi_I u_I\right)^T m \left(\sum_{J=1}^{n} \Phi_J \ddot{u}_J\right) d\Omega + \int_\Omega \delta \left(\sum_{I=1}^{n} \Phi_I u_I\right)^T c \left(\sum_{I=1}^{n} \Phi_I \dot{u}_I\right) d\Omega +$$
$$\int_\Omega (L \left(\sum_{I=1}^{n} \Phi_I \delta u_I\right))^T (DL \left(\sum_{J=1}^{n} \Phi_J u_J\right)) d\Omega - \int_\Omega \left(\sum_{I=1}^{n} \Phi_I \delta u_I\right)^T b d\Omega -$$
$$\int_{\Gamma_t} \left(\sum_{I=1}^{n} \Phi_I \delta u_I\right)^T \bar{t} d\Gamma + \int_{\Gamma_u} \left(\sum_{I=1}^{n} \Phi_I \delta u_I\right)^T \alpha \left(\left(\sum_{J=1}^{n} \Phi_J u_J\right) - \bar{u}\right) \bar{t} d\Gamma = 0 \tag{6.98}$$

and thus

$$\int_\Omega \left(\sum_{I=1}^{n} \Phi_I \delta u_I\right)^T m \left(\sum_{J=1}^{n} \Phi_J \ddot{u}_J\right) d\Omega + \int_\Omega \left(\sum_{I=1}^{n} \Phi_I \delta u_I\right)^T c \left(\sum_{I=1}^{n} \Phi_I \dot{u}_I\right) d\Omega +$$
$$\int_\Omega (\left(\sum_{I=1}^{n} B_I \delta u_I\right))^T (\left(\sum_{J=1}^{n} DB_J u_J\right)) d\Omega - \int_\Omega \left(\sum_{I=1}^{n} \Phi_I \delta u_I\right)^T b d\Omega -$$
$$\int_{\Gamma_t} \left(\sum_{I=1}^{n} \Phi_I \delta u_I\right)^T \bar{t} d\Gamma + \int_{\Gamma_u} \left(\sum_{I=1}^{n} \Phi_I \delta u_I\right)^T \alpha \left(\left(\sum_{J=1}^{n} \Phi_J u_J\right) - \bar{u}\right) \bar{t} d\Gamma = 0 \tag{6.99}$$

Using the linearity of operators and the fact that the equation holds for any value of δu, we obtain the system

$$\boldsymbol{M\ddot{U}(t) + C\dot{U}(t) + (K + K^{\alpha})U(t) = F(t) + F^{\alpha}(t)} \tag{6.100}$$

with K the global stiffness matrix, M the global mass matrix, C the damping matrix and F the global force vector.

$$M_{IJ} = \int_\Omega \Phi_I^T \cdot m \cdot \Phi_J d\Omega \tag{6.101}$$

$$C_{IJ} = \int_\Omega \Phi_I^T \cdot c \cdot \Phi_J d\Omega \tag{6.102}$$

$$K_{IJ} = \int_\Omega B_I^T \cdot D \cdot B_J d\Omega \tag{6.103}$$

$$K_{IJ}^{\alpha} = \int_{\Gamma_u} \Phi_I^T \cdot \alpha \cdot \Phi_J d\Gamma \tag{6.104}$$

With α penalty factor matrix.

$$F_I = \int_\Omega \Phi_I^T \cdot b d\Omega + \int_{\Gamma_t} \Phi_I^T \cdot \bar{t} d\Gamma \tag{6.105}$$

and

$$F_I^{\alpha} = \int_{\Gamma_u} \Phi_I^T \cdot \alpha \cdot \bar{u} d\Gamma \tag{6.106}$$

K^{α} and F^{α} are the added terms for the stiffness matrix and force vector to enforce the essential boundary condition through the penalty method. $\boldsymbol{\ddot{U}}$, $\boldsymbol{\dot{U}}$ and $\boldsymbol{U}$ are the global acceleration, velocity and displacement vectors that collect the nodal values.

$$\boldsymbol{\ddot{U}} = \left\{\ddot{u}_1(t), \ddot{u}_2(t), \ldots, \ddot{U}_N(t)\right\}^T \tag{6.107}$$

$$\dot{\boldsymbol{U}} = \left\{ \dot{u}_1(t), \dot{u}_2(t), \ldots, \dot{U}_N(t) \right\}^T \tag{6.108}$$

In order to lead the discretisation regarding time, the explicit or implicit scheme can be used. Explicit schemes are conditionally stable and the trade-off between accuracy, stability and computation time needs to be made. For this reason the implicit approach is preferred as it has an unconditional stability for large time steps. There are several well-known approaches such as Newmark-β, Wilson-θ or Runge-Kutta methods. We use the Newmark-β on the system of equation 6.100. Using Taylor series on the velocity $\dot{\boldsymbol{U}}$ and the displacement vector $\boldsymbol{U}$, we have

$$U_{t+\Delta t} = U_t + \Delta t \dot{U}_t + \frac{\Delta t}{2} \ddot{U}_t + \frac{\Delta t}{6} \dddot{U}_t + \ldots \tag{6.109}$$

$$\dot{U}_{t+\Delta t} = \dot{U}_t + \Delta t \ddot{U}_t + \frac{\Delta t}{2} \dddot{U}_t + \ldots \tag{6.110}$$

Using either the extended mean value (Cauchy) or the Newton truncated series allow to express

$$U_{t+\Delta t} = U_t + \Delta t \dot{U}_t + \frac{\Delta t}{2} \ddot{U}_t + \beta \Delta t^3 \dddot{U}_t \tag{6.111}$$

$$\dot{U}_{t+\Delta t} = \dot{U}_t + \Delta t \ddot{U}_t + \gamma \Delta t^2 \dddot{U}_t \tag{6.112}$$

Assuming linear variation of the acceleration component over Δt such as

$$\dddot{U}_t = \frac{\ddot{U}_{t+\Delta t} - \ddot{U}_t}{\Delta t} \tag{6.113}$$

gives

$$U_{t+\Delta t} = U_t + \Delta t \dot{U}_t + \left(\frac{1}{2} - \beta \right) \Delta t^2 \ddot{U}_t + \beta \Delta t^2 \ddot{U}_{t+\Delta t} \tag{6.114}$$

$$\dot{U}_{t+\Delta t} = \dot{U}_t + (1 - \gamma) \, \Delta t \ddot{U}_t + \gamma \Delta t \ddot{U}_{t+\Delta t} \tag{6.115}$$

The variables β and γ control the stability of the method as well as weighting of the damping effect into the main equation. From equation 6.114 we obtain

$$\ddot{U}_{t+\Delta t} = \frac{1}{\beta \Delta t^2} \left(U_{t+\Delta t} - U_t \right) - \frac{1}{\beta \Delta t} \dot{U}_t - \left(\frac{1}{2\beta} - 1 \right) \ddot{U}_t \tag{6.116}$$

Substituting equation 6.116 into 6.115 leads to

$$\dot{U}_{t+\Delta t} = \frac{\gamma}{\beta \Delta t} \left(U_{t+\Delta t} - U_t \right) + \left(1 - \frac{\gamma}{\beta} \right) \dot{U}_t + \left(1 - \frac{\gamma}{2\beta} \right) \Delta t \ddot{U}_t \tag{6.117}$$

Finally, substituting equation 6.116 and 6.117 within 6.100 gives

$$\begin{aligned} \left(\boldsymbol{M} \alpha_1 + \boldsymbol{C} \gamma \alpha_2 + \boldsymbol{K} \right) \boldsymbol{U_{t+\Delta t}} = \boldsymbol{M} \left(\alpha_1 \boldsymbol{U}_t + \alpha_2 \dot{\boldsymbol{U}}_t + \alpha_3 \ddot{\boldsymbol{U}}_t \right) + \\ \boldsymbol{C} \left[\gamma \alpha_2 \dot{\boldsymbol{U}} + \left(\frac{\gamma}{\beta} - 1 \right) \dot{\boldsymbol{U}}_{\boldsymbol{t}} + \left(\frac{\gamma}{2\beta} - 1 \right) \Delta t \ddot{\boldsymbol{U}}_{\boldsymbol{t}} \right] + \boldsymbol{F_{t+\Delta t}} \end{aligned} \tag{6.118}$$

with

$$\alpha_1 = \frac{1}{\beta \Delta t^2}; \quad \alpha_2 = \frac{1}{\beta \Delta t}; \quad \alpha_3 = \frac{1}{2\beta} - 1 \tag{6.119}$$

Thus we obtain the final system

$$\begin{aligned}(\boldsymbol{M}\alpha_1 + \boldsymbol{C}\gamma\alpha_2 + \boldsymbol{K})\,\boldsymbol{U_{t+\Delta t}} = \boldsymbol{M}\left(\alpha_1\boldsymbol{U}_t + \alpha_2\dot{\boldsymbol{U}}_t + \alpha_3\ddot{\boldsymbol{U}}_t\right) + \\ \boldsymbol{C}\left[\gamma\alpha_2\dot{\boldsymbol{U}} + \left(\frac{\gamma}{\beta} - 1\right)\dot{\boldsymbol{U}}_{\boldsymbol{t}} + \left(\frac{\gamma}{2\beta} - 1\right)\Delta t\ddot{\boldsymbol{U}}_{\boldsymbol{t}}\right] + \boldsymbol{F_{t+\Delta t}}\end{aligned} \tag{6.120}$$

with

$$\alpha_1 = \frac{1}{\beta\Delta t^2}; \quad \alpha_2 = \frac{1}{\beta\Delta t}; \quad \alpha_3 = \frac{1}{2\beta} - 1 \tag{6.121}$$

The choice of parameters β and γ leads to different versions of the Newmark methods. The method is conditionally stable for $\beta \geq \frac{\gamma}{2} \geq \frac{1}{4}$. When $\gamma < 0.5$, damping is introduced. For $\gamma = 0.5$ $\beta = \frac{1}{4}$ or $\beta = \frac{1}{6}$, it leads to the average acceleration and constant acceleration method respectively. For $\gamma = \frac{3}{2}$ and $\beta = \frac{8}{5}$ we obtain the Galerkin method.

6.6.1 Natural Frequency (Modal) analysis

All things vibrate, from audible sounds coming from musical instruments, cars, airplanes, buildings, to the subtlest vibration of microstructures such as nano-tube or quartz lattices. Vibration is not ideal and often unavoidable. It can cause a gradual weakening of structures and the deterioration of metals components. The most disastrous consequences occur when a power-driven device, such as a motor for example, produces a frequency at which an attached structure naturally vibrates. This event is called resonance. When a vibration causes resonance in an object, destruction will result unless it has been designed to withstand the stress. Engineers must design objects such that resonance does not occur during the regular operation of machines. This is a major objective of natural frequency (modal) analysis. Ideally, the first mode has a frequency higher than any potential driving frequency.

Modal analysis requires more degrees of freedom in the model than the number of frequencies (modes) being calculated. Thus, simplified "test" models may not provide meaningful result if the discretised model is very coarse. Many MEMS sensors such as the gyroscope or accelerometer use the principle of vibration, thus for this reason this type of problems constitutes a good benchmark. In modal analysis, in order to obtain the natural vibration frequencies we consider the system equation 6.100 with no damping and body forces.

$$M\ddot{U} + KU = 0 \tag{6.122}$$

The displacement is assumed to be a harmonic function of time

$$u(x,t) = \begin{bmatrix} u'(x)\sin(wt+\phi) \\ v'(x)\sin(wt+\phi) \end{bmatrix} \tag{6.123}$$

As usual, using the meshfree approximation at any nodes x, we can write

$$u(x,t) = \sum_{k=1}^{n} \begin{pmatrix} \phi_k & 0 \\ 0 & \phi_k \end{pmatrix} \begin{Bmatrix} u'_k \\ v'_k \end{Bmatrix} \sin(wt+\phi) = \Phi(x)U'\sin(wt+\phi) \tag{6.124}$$

With n the number of nodes within the support domain of the shape function at the sampling point. w is natural pulsation and ϕ the phase. Vector U' contains the displacement components in x and

y directions. We substitute equation 6.124 into the 6.122 and simplify the expression to obtain

$$(K - w^2 M)U' = 0 \tag{6.125}$$

We assume $\lambda = w^2$ to obtain the final equivalent eigenvalues equation system

$$(K - \lambda M)q = 0 \tag{6.126}$$

λ are the eigenvalues of the system and q the corresponding eigenvectors. Solving the eigenvalues system 6.126 gives the i=1...N frequencies such that $f_i = w_i/2\pi$ and their associated modes which correspond to the eigenvectors q_i We consider the same beam system studied in section5.5 page.114. The mass density of the beam material is $\rho = 1kg.m^{-3}$.
A comparison of natural frequencies for the first 10 modes using FEM, EFG with MLS and RPIM shape functions is given in table 6.1. For the MLS and RPIM, we use an irregular distribution of 231 nodes (21×11) and $\alpha_s = 2.5$. For FEM a regular distribution with QUAD element is used. The perturbation parameter for the irregular distribution of nodes is set to 3.5 (see equation 7.21 p.173). Once again both FEM and meshfree methods algorithm are tested using our own developed Matlab code. The reference solution correspond to a FEM solution with 10K DOFs. Figure 6.45 shows the displacement modes obtained with regular distribution FEM with Quad element and EFG using irregular distribution of nodes. They are plotted using our Matlab script.

	Natural mode frequencies (Hz)			
Modes	ref	FEM 19x9	EFG MLS	EFG RPIM
1	4,412	4.457	4.359	4.339
2	22,419	22.739	22.2142	22.209
3	28,602	28.6246	28.416	28.348
4	51,549	52.6026	51.146	51.240
5	83,306	85.68	82.682	82.893
6	85,398	85.71	84.817	84.674
7	116,189	120.59	115.241	115.374
8	140,260	141.466	139.247	139.191
9	143,190	147.017	142.295	142.120
10	159,189	163.701	158.221	157.800

Table 6.1: Natural frequencies for first 10 modes for beam free vibration.

The results on table 6.1 show a good agreement between MF method's results and reference solutions even with irregular node distribution. Both RPIM and EFG perform better than the regular QUAD bilinear FEM element.

6.6.2 Forced vibration analysis

We now study the same beam system but under dynamics load. The problem is the same as in section 5.5, but this time the parabolic load at the free end is a function of the time. For this problem we use the Newmark method for the time integration and the final system of equation is given by 6.6. For this problem we compare the solution of weak formulated meshfree methods with the one obtained with our FEM code on Matlab, using 10×21 QUADs bilinear element. We first consider a dynamic relaxation problem with a constant parabolic load. Figure 6.46 shows the

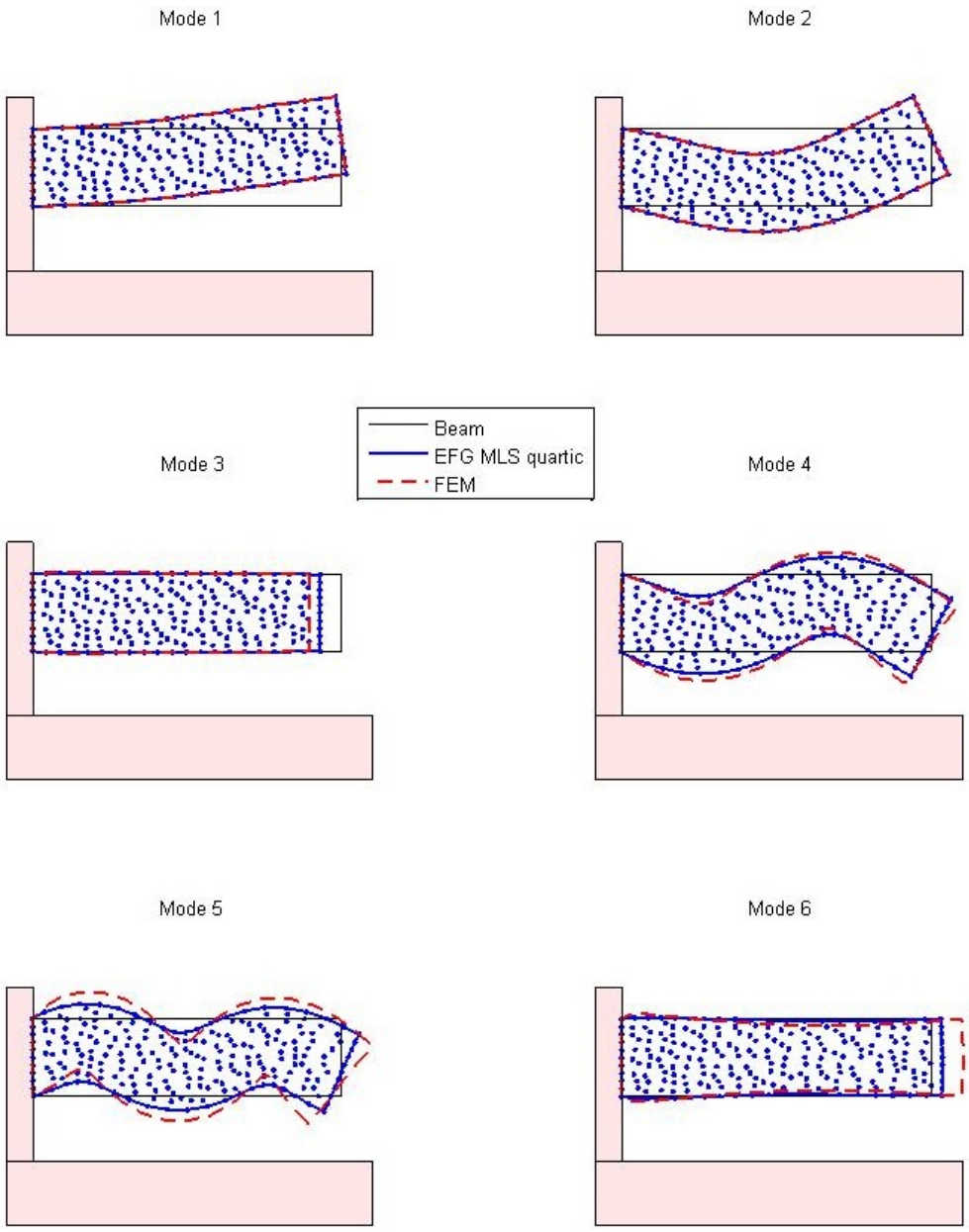


Figure 6.45: 2D cantilever beam free vibration modes using FEM and EFG

displacement of the beam at the middle point of the free end under a step load. The solution is obtained with a 11×5 regular node distribution. Linear basis is used for MLS shape function with a regular distribution of nodes of 10×21, and $\alpha_s = 2.5$. As expected, the solution reaches the equilibrium position of the elastic beam problem of section5.5. Figure 6.47 shows a comparison of the displacement through time obtained using EFG (blue) and FEM (black). Figure 6.49 shows a close up comparison of the EFG , FEM and analytical solutions around the mid end tip of the beam. Figure 6.48 illustrates how both methods converge slightly differently to the final displacement. We can observe that FEM rather overestimated the solution whereas the EFG undervalued it through the final step of the iteration process. As we can, see on figure 6.49 the solution converges very well to the exact solution. The errors for FEM and EFG are 2.3% and 0.7% respectively. This again illustrates the capability of EFG to be more accurate for the same DOF.

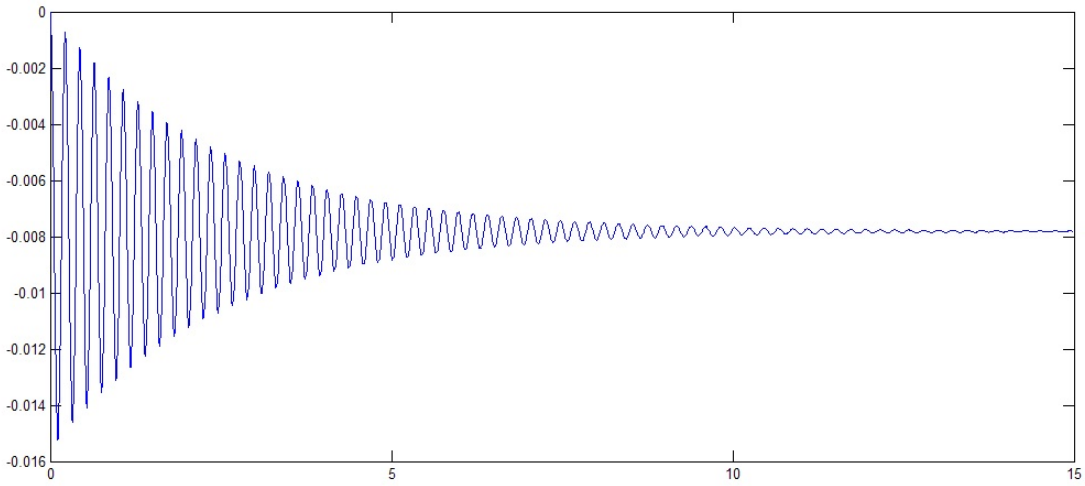


Figure 6.46: 2D cantilever beam displacement a free end under step load obtained with EFG. 21×11 regular node distribution. $\alpha_s = 1.5$.

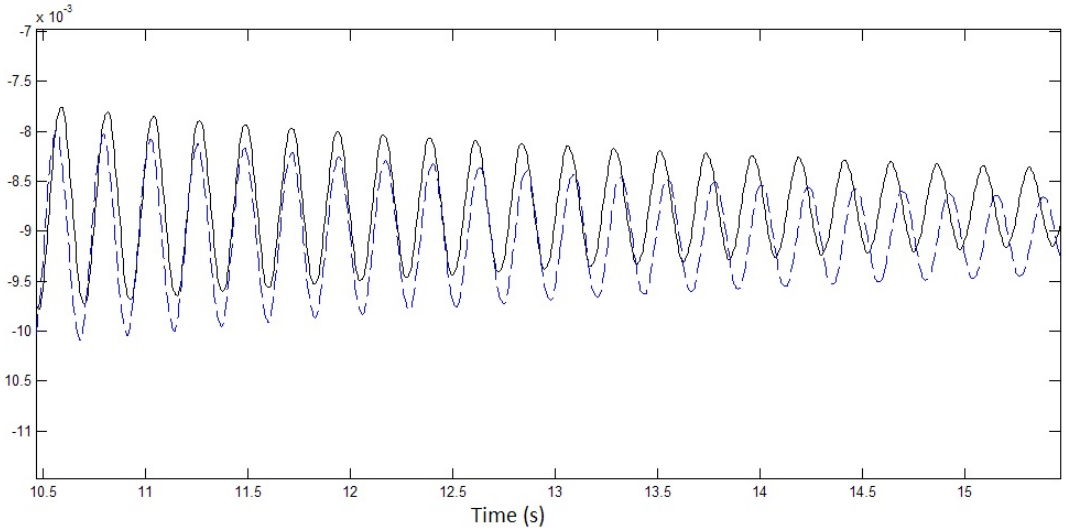


Figure 6.47: 2D cantilever beam displacement a free end under step load obtained with EFG (blue) and FEM (black). 11×5 regular node distribution. $\alpha_s = 2.5$.

We now study the response of the beam to the following linear decreasing loading $P(t)$

$$P(t) = \begin{cases} 1000(1-t); & \forall t \in [0,1[\\ 0; & \forall t > 1 \end{cases} \tag{6.127}$$

Figure 6.50 shows the response of the beam. Same parameters as for the previous case are used. On figure 6.50, we can observe that the response of FEM and EFG is similar. The oscillation appeared to be out of phase.

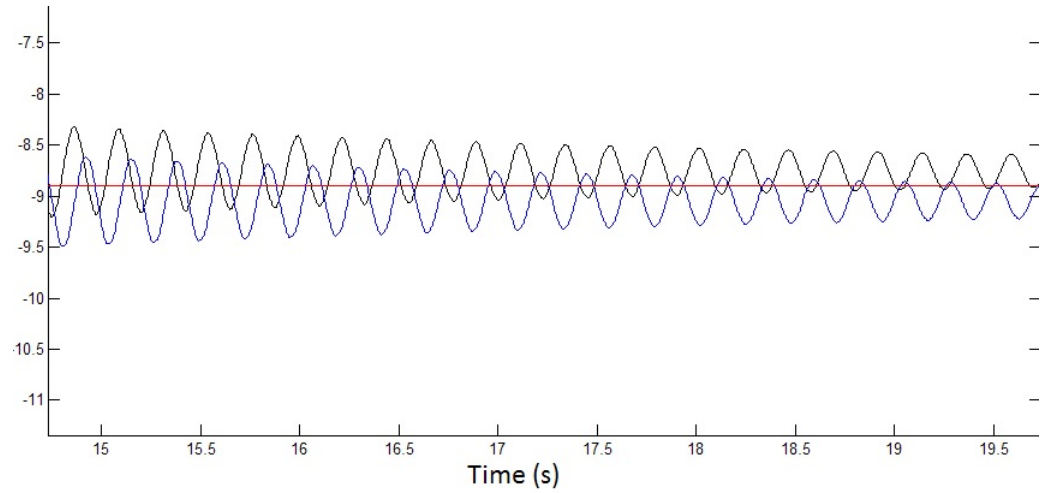


Figure 6.48: Convergence to the solution displacement(red) at the free end mid-point under step load, obtained with EFG (blue) and FEM (black).

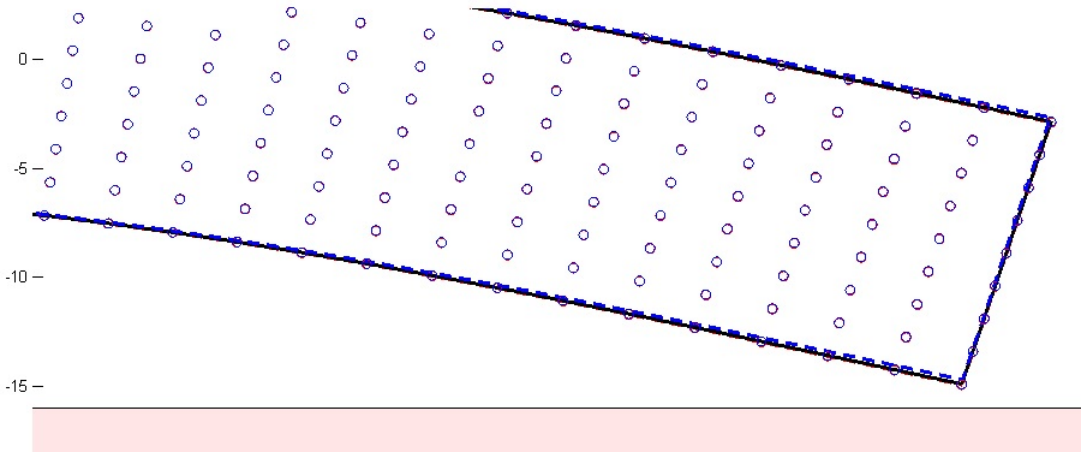


Figure 6.49: 2D cantilever beam displacement a free end under step load obtained with EFG (blue) and FEM (black). 11×5 regular node distribution. $\alpha_s = 1.5$.

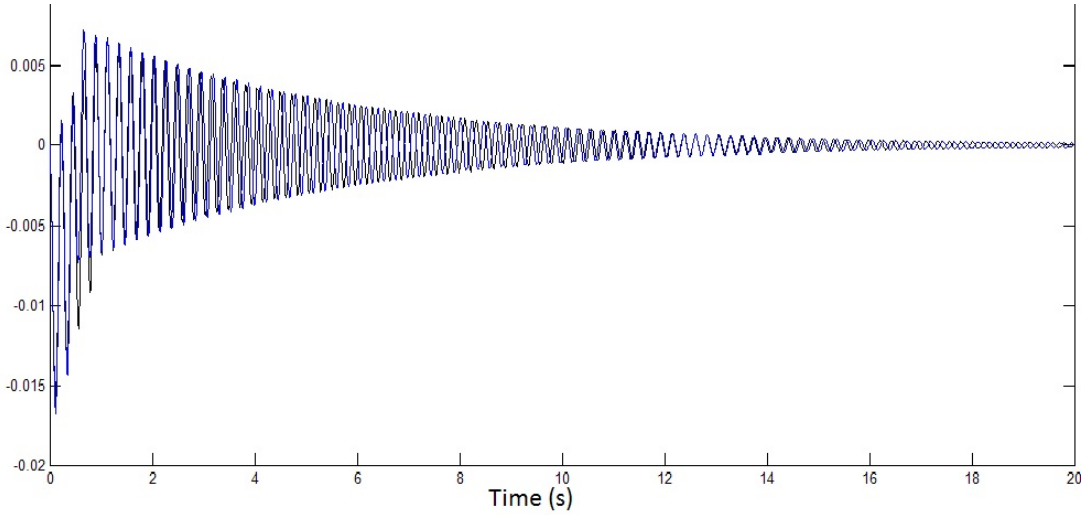


Figure 6.50: 2D cantilever beam displacement a free end under pulse load obtained with EFG (blue) and FEM (black). 11×5 regular node distribution. $\alpha_s = 2.5$.

Figure 6.51 shows a comparison for EFG with MLS linear basis for a 242 regular and irregular($\alpha_{ifc} =$

0.4) distribution of nodes. As shown, the method is robust for irregular node pattern. Similar re-

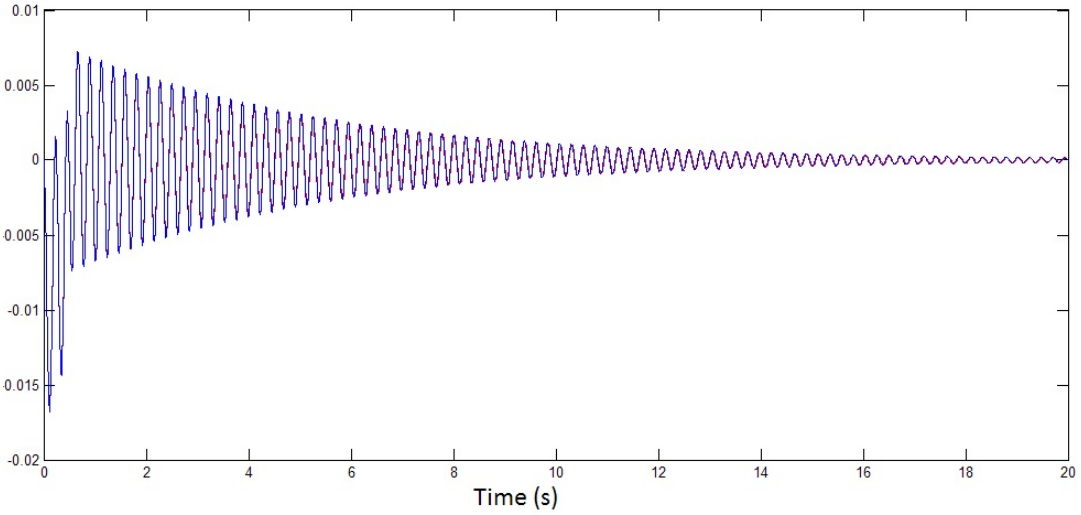


Figure 6.51: 2D cantilever beam displacement a free end under pulse load obtained for regular (blue) and irregular distribution of nodes (red)

sults are displayed on figure 6.52 for the RPIM-WEN C6 SF.

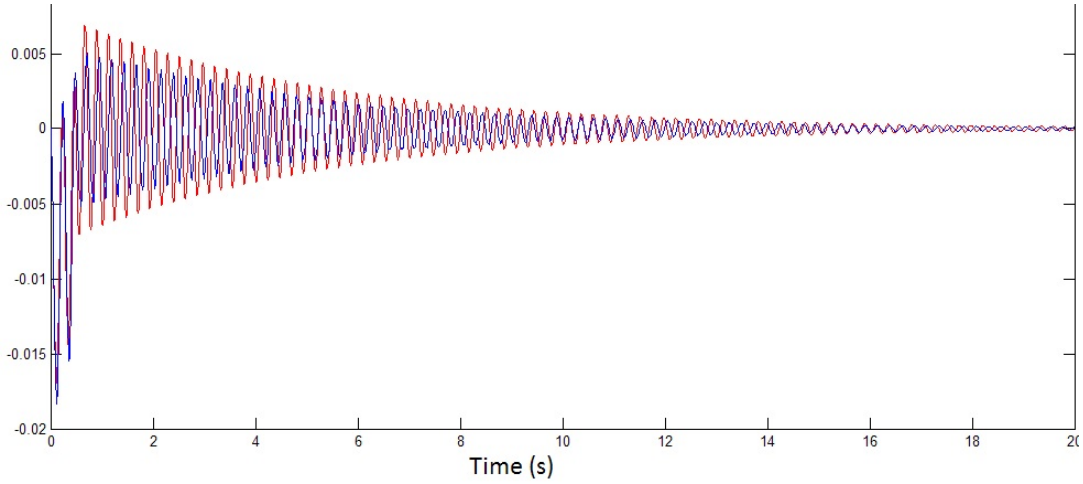


Figure 6.52: 2D cantilever beam displacement a free end under pulse load using RPIM-WEN C6. obtained for regular (blue) and irregular distribution of nodes (red)

6.7 Conclusion

In this chapter, we focused on the implementation of EFG for benchmark problems for potential haptic MEMS simulation applications. We covered thermal oriented problems in 2D and 3D. We also covered elastic-dynamic deformation of a beam model which is a regular component in thermal based sensor and other MEMS structures. This allowed us to further highlight the pitfalls and drawbacks in the implementation of the methods.

Overall, the adaptive constellation of nodes compared to the rigid mesh used in conventional FEM provides flexibility and the possibility to conform to complex geometries even in 3D. The generation of the constellation of nodes is computationally inexpensive compared to the building of a mesh. Methods using random or quasi random such as Niederreiter [203] or Sobol [265] algorithm to minimise the clustering effect, can be used. However the inherent absence of constraints

also comes with an absence of support for the data structure that usually provides the mesh. The computation of the influence domain and the process of finding the nodes belonging to the support domain for a point of interest ends up being computationally expensive as the number of field nodes used to discretise the domain grows. On the other hand, the use of hierarchical structures and geometry in mesh methods is highly beneficial for such procedures.
Due to the node based approximation and absence of mesh-link geometry, discontinuities in the geometry, laminated components materials and nonlinearities require special treatments. These problems are not as straightforward as they usually are in FEM, where the support of elements for describing the physical properties through the geometry provides a comfortable solution.
The h-refinement is not facilitated by the fact that it is easy to add node to the original constellation without any impact on the pre-existing state of problem data. Again, the absence of mesh results in the lack of a structure to perform the procedure. Nonetheless, in EFG, grid to support the integration can be used to this extent along with bucket algorithm.
Another disadvantage of MF methods due to the absence of mesh, lies in the use of lists and search algorithms among dynamic list of nodes. There are many tasks which require the use of such list and search processes. In order to control the computational cost of MF algorithm, attention needs to be paid to the solution to deal with each of these lists. This can be accomplished with dedicated automated algorithms (bucket algorithm, Delaunay tessellation). Because the size of these lists remain relatively compact, it does help to treat them efficiently. Figure 6.2 provides a comparison of the features of EFG methods with FEM. Paradoxically, if MF based on weak form methods never

Characteristics	**FEM**	**Weak form Meshfree (EFG)**
Mesh **Shape functions**	Needed Based on pre-defined elements	Grid for integration Based on local nodes over support domains
Discretised system **Stiffness matrix**	Banded, symmetric	Banded, depending on the method used, not symmetric
Imposition of **Essential Bcs**	standard procedure for FEM. Easy	Require special treatment for many meshfree SF
Computation speed	Linked to DOFs and nb of quadrature points	More costly than FEM, Slower for MLPG
Accuracy/ **Adaptive analysis**	More accurate than FDM complicated for complex 3D	More accurate than standard FEM. / easy
Stage of development	Mature technology	Infant
Commercial software	Many	Few

Table 6.2: Comparison of FEM and EFG features

rely on mesh for the building of their approximation functions, some geometric data structures or support for particular steps in the computation of the solution, such as integration or node search, is required. Hence it relies on the use of meshing tools such as Delaunay triangulations, Voronoi diagram or cell-tree. Due to the impact of integration on the overall computation, we study the collocation form of meshfree in the next chapter, as a way to mitigate the computational cost.

Chapter 7 Meshree methods based on strong formulation: Collocation Methods

In the previous Chapter we studied and applied the meshfree SFs within the weak formulation framework. A huge part of the computational cost of these methods is dedicated to the integration requirements. For MLPG, the management of subdomains and intersections with boundaries is a costly task. Integration over subdivisions of the local domain is used to ensure the compatibility but this will increase the computational cost. EFG on the other hand, is faster but relies on the use either of background grid or Voronoi cell in the case of SCNI. If the first one does not need to conform to the boundary or shape of the domain, there is still some extra computation charge dedicated to the integration of the SFs. A fine grid with enough Gauss points is necessary for the accurate integration of non-polynomial form of meshfree SFs. In the case of SCNI, the Voronoi cell is built over a cell. Despite of the increased accuracy in the gradient of the field solution, this still requires a mesh. Hence it defies our purpose i.e methods that can remove the need to build the physics model over a mesh and all the related re-meshing procedures. For this reason, we investigate meshfree methods based on Collocation methods and compare them with weak formulated ones.

7.1 Collocation method

Methods based on the strong formulation have been studied intensively in the past. One of the most famous is the finite difference method (FDM) which since the early 70's have been use in many areas to solve engineering problems. In order to approximate strong-formulations of PDE and obtain a set of linear equations, a collocation method is applied at the field nodes which serve as sampling locations.
If we consider the residual formulation of a PDE over a domain Ω (appendix A p.204):

$$\int_{\Omega} RW_i d\Omega \quad i = 1, ..., M \tag{7.1}$$

The idea of the collocation method also called point collocation method is to satisfy the PDE equation only at certain "points" distributed over the domain Ω. In the collocation method the Dirac delta function is used as the weight function

$$W_i = \delta(x - x_i) \tag{7.2}$$

Knowing that

$$\int_{\Omega} f(x)\delta(x - x_i)d\Omega = f(x_i) \tag{7.3}$$

The collocation process produces a set of discrete equations (one for each collocation points) of

the form:

$$R(u_1, ...u_M, x_i) = 0 \quad i = 1, ..., M \tag{7.4}$$

Where u_i being the collocation points at which the partial differential equation is now required to be satisfied by forcing the residual to vanish at this location. Here we can see how the collocation method meets the Finite difference method in their formulation. The FDM is simply a collocation where the derivative at each collocation point is replaced by a finite difference scheme.
The system of equations A.5 p.204 with the collocation method results in the discrete equations which enforce the residual to vanish at M distinct collocation points x_i

$$R_s(\hat{u}(x_i)) + R_b(\hat{u}(x_i)) = 0 \tag{7.5}$$

Point collocation method is the starting point for numerical techniques such as the finite difference method (FDM). The finite difference is simply the collocation method with the derivatives at each collocation point formulated by finite difference approximations. In this case the basis function is locally defined.
Many meshfree strong-form methods have been developed so far. They are more or less the same collocation method used with different type of shape functions. We can cite the Hp-meshless cloud method [157, 158], the finite point method(FPM) [173, 207–210], the Vortex method, which mainly is dedicated to solve engineering flow problems [60, 61, 136, 145, 152, 271], the RPKM point collocation method [8]. The general finite difference method (GDFM) and its recent special form like the multiquadric finite difference method [257] are other methods as well. Finally the meshfree collocation method [132, 133, 171, 172] using RPIM SFs. Liu also apply the collocation framework with PIM SFs [171, 172].

7.2 Meshfree collocation implementation procedure and issues

We consider the following 2D Poisson problem governed by the strong formulation

$$\nabla^2 u + u = (2 + 3x)e^{x-y} \quad \forall x_I = (x, y) \in [0, 1] \times [0, 1] \tag{7.6}$$

with the following essential boundary conditions

$$u(0, y) = 0; \quad \forall x_I = (x, y) \in \Gamma_{u1} \tag{7.7}$$

$$u(x, 0) = xe^x; \quad \forall x_I = (x, y) \in \Gamma_{u2} \tag{7.8}$$

The derivative boundary conditions (DBCs) are given

$$\frac{\partial u(1, y)}{\partial x} = e^{1-y}; \quad \forall x_I = (x, y) \in \Gamma_{db1} \tag{7.9}$$

$$\frac{\partial u(x, 1)}{\partial y} = xe^{x-1}; \quad \forall x_I = (x, y) \in \Gamma_{db2} \tag{7.10}$$

Using meshfree SFs (section 4.2 p.59), the unknown field function solution is expressed at a collocation point x_I

$$u^h(x_I) = \Phi_I^T u_I \tag{7.11}$$

and its derivative

$$\frac{\partial u^h(x_I)}{\partial x} = \frac{\partial \Phi_I^T}{\partial x} u_I; \quad \frac{\partial^2 u^h(x_I)}{\partial x^2} = \frac{\partial^2 \Phi_I^T}{\partial x^2} u_I; \quad \frac{\partial u^h(x_I)}{\partial y} = \frac{\partial \Phi_I^T}{\partial y} u_I; \tag{7.12}$$

$$\frac{\partial^2 u^h(x_I)}{\partial x \partial y} = \frac{\partial^2 \Phi_I^T}{\partial x \partial y} u_I; \quad \frac{\partial^2 u^h(x_I)}{\partial y^2} = \frac{\partial^2 \Phi_I^T}{\partial y^2} u_I \tag{7.13}$$

As usual Φ_I is the shape functions vector at x_I and u_I the vector of the nodal values of the unknown field function. Hence, using collocation method, we obtain the following discretised system of equation for node x_I

$$\left[\frac{\partial^2 \Phi_I^T}{\partial x^2} + \frac{\partial^2 \Phi_I^T}{\partial y^2} + \Phi_I^T \right] u_I = (2 + 3x_I) e^{x_I - y_I} \tag{7.14}$$

Applying the equation for each node x_I of the domain leads to the regular matrix system

$$K_I u = f_I \tag{7.15}$$

For the N_u nodes $x_I \in \Gamma_u$ We have the following independent equations

$$\Phi_I^T u_I = 0; \quad \forall x_I = (x, y) \in \Gamma_{u1} \tag{7.16}$$

$$\Phi_I^T u_I = x e^x \quad \forall x_I = (x, y) \in \Gamma_{u2} \tag{7.17}$$

We can enforce the following Dirichlet boundary conditions by direct interpolation (see section 5.2.4 p.110). For a node $x_I \in \Gamma_{u1}$ or Γ_{u2} we can replace respectively the I^{th} line of the system equation 7.15 by either the system equation 7.16 or 7.17. When the Sfs used possess the Kronecker delta property, it is even easier as for a nodes $x_I \in \Gamma_u$ the system matrix entry $K(I, I) = 1$ and $f_I = \bar{u_I}$ with $\bar{u}$ being the essential boundary condition values at nodes x_I.
For the derivatives boundary conditions (DBCs) at the N_{db} DB-nodes x_I, we have to enforce the following independent equations

$$\frac{\partial \Phi_I}{\partial x} u_I = e^{1-y}; \quad \forall x_I \in \Gamma_{db1} \tag{7.18}$$

$$\frac{\partial \Phi_I}{\partial y} u_I = x e^{x-1}; \quad \forall x_I \in \Gamma_{db2} \tag{7.19}$$

This can be done by replacing the I^{th} line in the system of equation 7.15 by one of the two equations 7.18 or 7.19. However, this does not enforce the DBCs properly at the nodes x_I. We will highlight this issue in the following section.
The code used for all our tests in this chapter has been developed using Matlab and is available for further development. We use the following norms as the error indicator in our tests

$$e = \sqrt{\frac{\sum_{i=1}^{N} \left(u_i^e - u_i^h \right)}{\sum_{i=1}^{N} \left(u_i^e \right)}} \tag{7.20}$$

Where u_i^e is the nodal exact value and u_i^h the approximated values using the collocation method. In order to control the irregularity of the nodal distribution, we use Irregular factor (Ifc) k to control

the magnitude of perturbations for the different nodes distribution such as

$$x = x_{reg} + d_x \times k \times \alpha_{rand}; \quad y = y_{reg} + d_y \times k \times \alpha_{rand} \tag{7.21}$$

Where $(x, y)_{reg}$ are coordinates for a regular distribution of 441 nodes, d_x and d_y the nodal spacing in x and y directions and the random number $\alpha_{rand} \in [-1; 1]$.
The issue with the collocation method is its tendency to be unstable. The problem lies in the implementation of the natural boundaries conditions or derivative boundaries conditions (DBCs). In order to illustrate this phenomenon, we study the problem governed by equation 7.6 along with its BCs. In order to compare the effect of improper treatment of the DBCS on the solution accuracy, we compare to the same problem but we replace the DBCs in equation 7.22 with the following essential BCs

$$u(1, y) = 2e^{1-y}; \quad u(x, 1) = -xe^{x-1} \tag{7.22}$$

In both scenario the analytical solution of the problem is given by

$$u_e(x, y) = xe^{x-y} \tag{7.23}$$

We use 5×5, 11×11, 21×21 and 31×31 regular distributions of nodes. We use RPIM SFs MQ with $\alpha_c = 3$ and $Q = 1.03$, EXP with $\alpha_c = 0.03$ and WEN4 wit $\delta = 50$ as described in section 4.3 p. 74. Figure 7.1 shows a comparison of the convergence of the collocation methods with and without DBCs for a regular distribution of nodes.

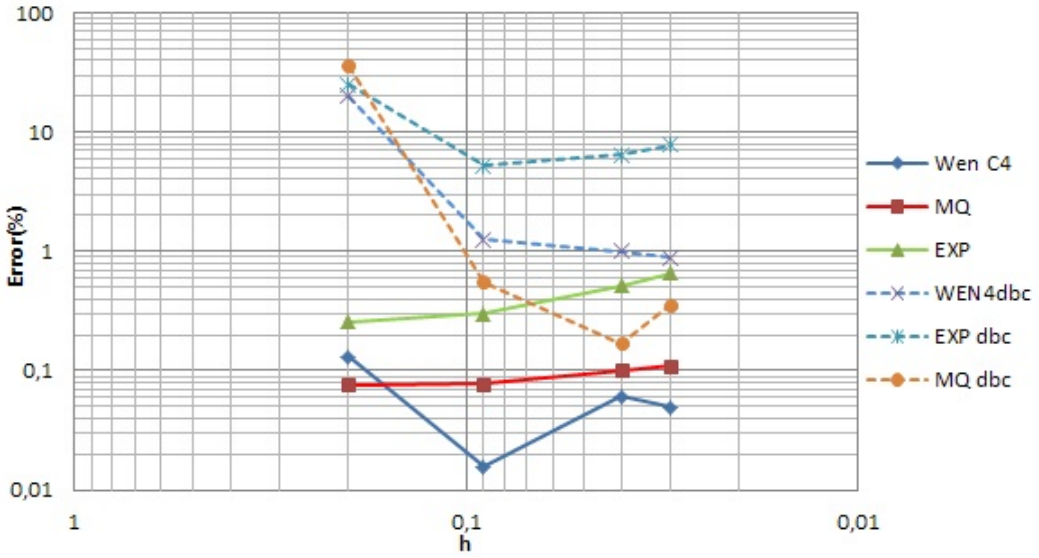


Figure 7.1: Convergence of RPIM collocation methods with and without DBC for regular distribution of nodes. $\alpha_s = 2$

It is clear that the presence of DBC seriously affects the accuracy of the method and its convergence. A similar study is done with a irregular distribution of nodes. Figure7.2 shows the results. In the case of nodes irregularly distributed, the situation is worse and the instability of the methods is greater. The difference between the case when there is the presence of DBCs and the one with only Dirichlet conditions (DC) is more important.
Figure 7.3 and 7.4 shows that, as the density of nodes increases, better results are achieved by keeping the size of the support domain to a certain relative size to the characteristic length of the

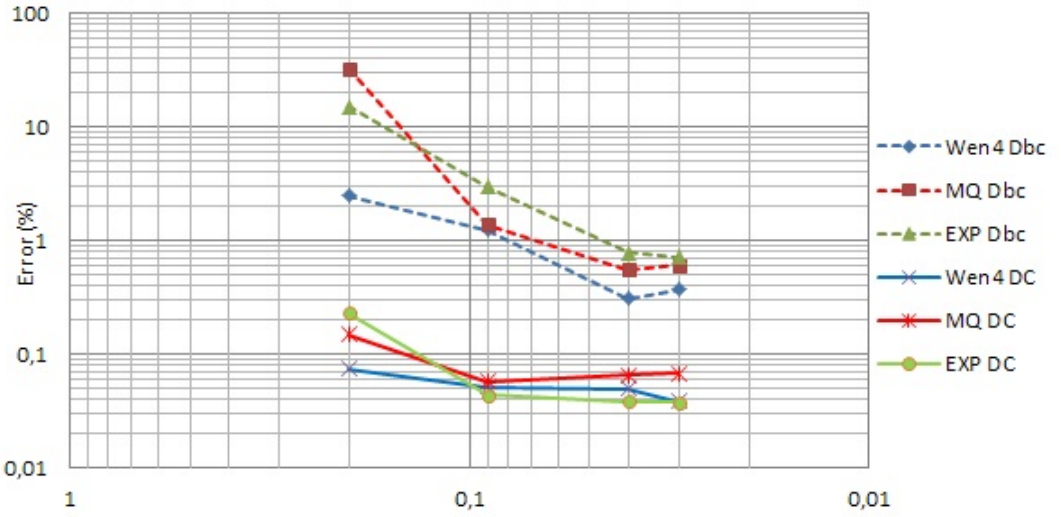


Figure 7.2: Convergence of RPIM collocation methods with and without DBC for irregular node distribution. $\alpha_s = 2$

domain. This can be done through the increase of the dimensionless parameter α_s.

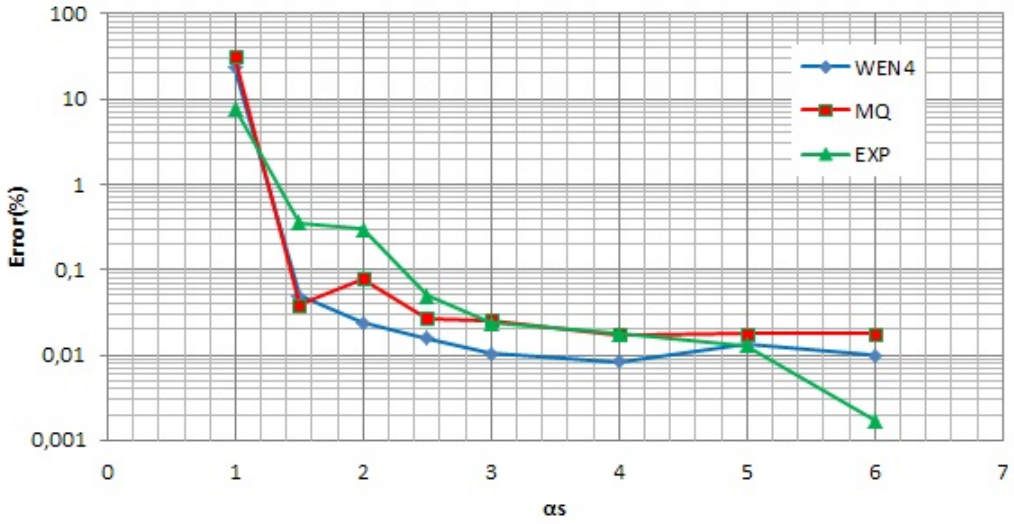


Figure 7.3: Influence of α_s on error of RPIM collocation. Regular distribution of 11×11 nodes.

In our case a range of $\alpha_s \in [3, 4]$ seems optimal which confirms the results of our preliminary studies of section 4.3 p. 79. Thus figure 7.5 shows the results using $\alpha_s = 3.5$ for the simulation of the case with only DC.

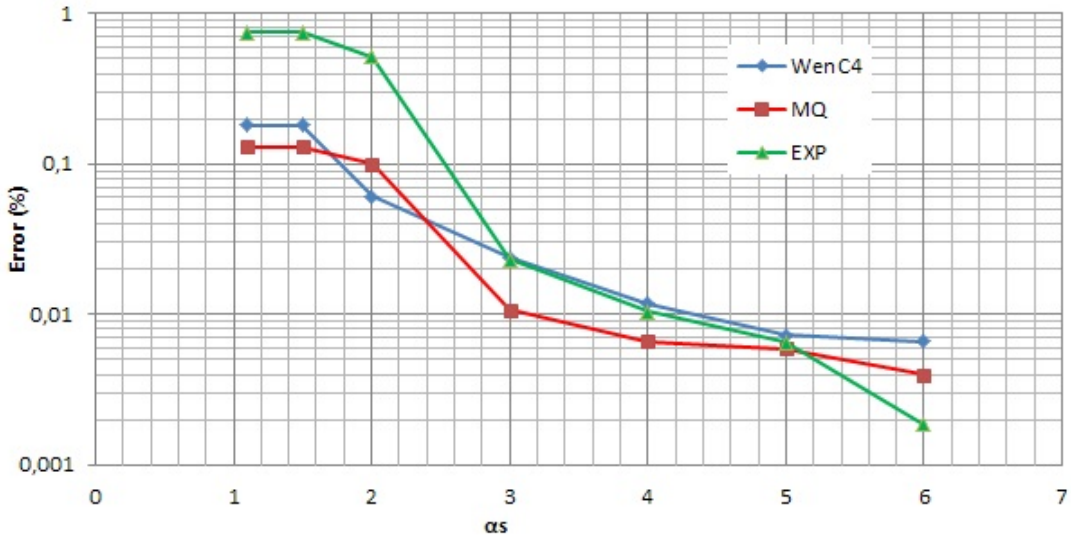


Figure 7.4: Influence of α_s on error of RPIM collocation. Regular distribution of 21×21 nodes.

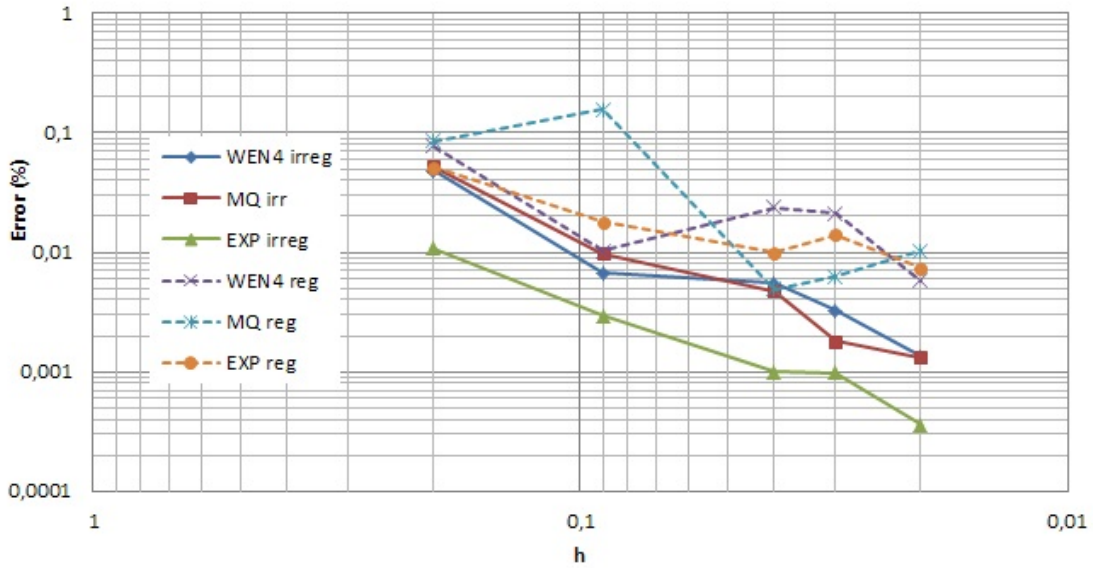


Figure 7.5: Convergence of RPIM collocation methods without DBC for regular and irregular node distributions.$\alpha_s = 3.5$

Figure 7.5 shows now a much similar convergence for regular and irregular distributions is obtained with $\alpha_s = 3.5$. Figure 7.6 gives the results obtain for the formulation with the DBC.
Comparing figure 7.5 and 7.6 still shows that presence of DBC results in a decrease of accuracy. It also results in sensitiveness to the irregularity of the node distribution with rather unstable results. To illustrate this, we study the standard deviation σ of the error for different irregular 441 nodes distributions. Here, we consider irregular factor Ifc of 0.3 and 0.6. σ is computed over more than 10 samples error values for each set of parameters. Tab 7.1 shows the results.
From tab 7.1, it shows that an increase in the irregularity of nodes distribution affects methods with DBCs more. For DC boundaries, an increase in support domain size through α_s helps to reduce the sensitivity of the methods to the irregularity of nodes distributions. For $\alpha_s = 3.5$ the method's error dispersion is in order of the error obtained for regular nodes distribution. In the case of presence of DBC, increasing the support domain does not improve the result much. Nonetheless,

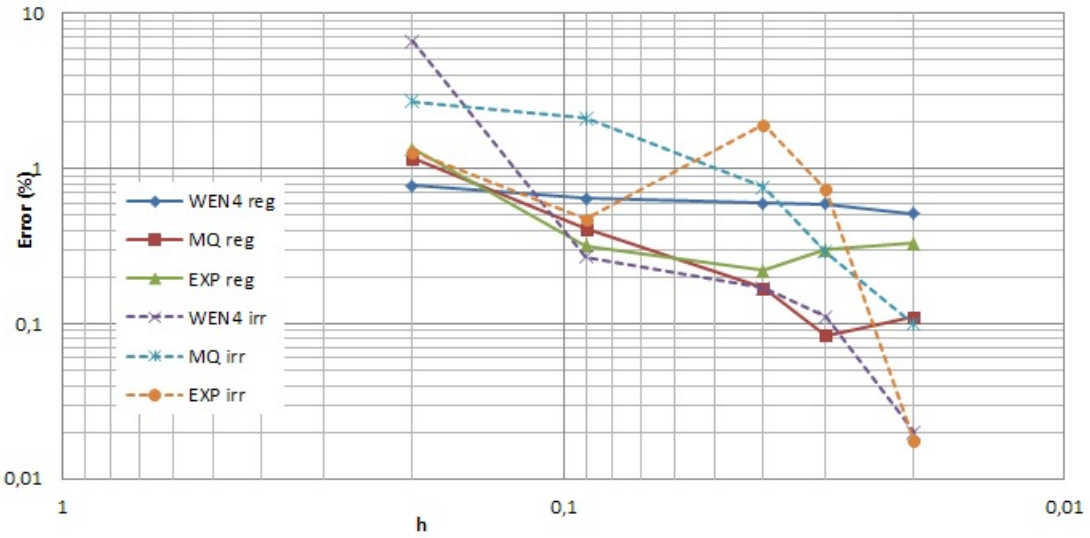


Figure 7.6: Convergence of RPIM collocation methods with DBC for regular and irregular node distributions.$\alpha_s = 3.5$

Irregularity factor		**Error for reg distrib** 0		**Standard deviation** σ 0.3		0.6	
alpha_s		2	3.5	2	3,5	2	3,5
DC	Wen4	6,08E-04	3,02E-05	8,38E-05	2,67E-05	0,000329	2,95E-05
	MQ	0,001	4,97E-05	0,000252	6,00E-06	0,000372	7,82E-06
	EXP	0,0052	9,89E-05	0,000409	4,75E-05	0,001541	2,38E-05
DBC	Wen4	0,0101	0,006	0,007655	0,020572	0,066495	0,081581
	MQ	0,0017	0,0017	0,033158	0,000537	0,011614	0,410756
	EXP	0,0648	0,0167	0,148696	0,006457	0,080262	0,005542

Table 7.1: Comparison of standard deviation for irregular distributions of nodes with different irregularity factors and for different α_s

failure to impose DBC correctly is responsible for a loss of accuracy in all cases.

7.3 Derivative boundary condition(DBC) treatment

Several techniques [162] have been developed to solve the problem:

- Adding fictitious points along the derivative boundary. Along the derivative boundaries, a set of fictitious points is added on the outer region surrounding the derivative boundary domain. This leads to the establishment of two sets of equations for each derivative boundary node(DB-node). There is one for the DBC and a second one for the governing system equation. This method is termed fictitious point(FP) method.

- Use of additional derivatives variables for the DB-nodes to enforce the DBCs. The derivatives of the function are added in the basis. This methods are referred as Hermite interpolation [159].

- Increasing the density of node along the derivative boundaries. An example was introduced by the work of Lizka for his Hp-cloud method [158].

- Using local weak formulation at the DBS nodes in order to use the natural way of implementing derivatives boundary condition. This technique is called mixed weak-strong (MWS) form [160,163]. This method is a combination of the local weak-form(see section5.2 p.104) and the strong-form. The local weak form allows to naturally satisfy the DBCs.

7.3.1 Fictitious points (FP)

The fictitious points method or FP method consists in augmenting the DOFs by adding along the derivative boundaries, a set of imaginary points. These fictitious points are added on the outer region surrounding the derivative boundary domain. For each DBC nodes, there are now two sets of equations. One is used for the DBC and a second one for the governing system equation. Through our test we found that a reasonable distance from the DBC for a fictitious point x_{fp} should be $d_{fp} = \lambda.n \cdot ds$ when n is the vector normal to the boundary and ds the vector of the characteristic length of the domain $ds^T = ds_x, ds_y$. ds_x and ds_y are the average nodal distance in x and y direction. For each DBC node x_I on the DBC, a FP is added ad the distance d_{fp} from the boundary. We found that best results are achieved for $\lambda \in [0.5, 1]$.
Using FP method on the Poisson problem governed by equation 7.6. For instance, we consider a 121 nodes distribution regularly dispatched as 11×11 over the square domain Ω. Fictitious point are dispatched along Γ_{db1} and Γ_{db2} such that there is $N_{FP} = 22$ added DOFS. For the N fields nodes x_I within the domain the system of equation is normally assembled using the procedure described in section 7.1. The only difference is that the shape function values at FPs are considered when they are involved in the support domain of a sampling point. Hence the original system matrix $(K_I)_{N\times N}$ becomes a $(K_I)_{N\times(N+N_{FP})}$. For each DBC point, the appropriate DBC equation is added in the system of equation at associated FPs line. N_{FP} lines are added at the system matrix K_I, hence giving the final system $(K_I)_{(N+N_{FP})\times(N+N_{FP})}$ incorporating the DBC conditions at each DB nodes.
Considering the Poisson problem of section 7.1 with DBC. We now control the number of nodes in the support domain. The number of nodes within the support domain for col-RPIM is kept constant at around 30-40 nodes, except for the compactly supported RBF, WEN-C4 where the number of nodes is slightly less. For WEN-C4, the best results have been obtained for a number around 20-25 nodes. For MLS SFs, the number is kept around 8-10. The IFC is set to 0.5. For RPIM-MQ, $\alpha_c = 3$ and for RPIM-EXP, $\alpha_c = 0.01$. It has been observed that, as the density of nodes increases, the number of nodes in the support domain (e.g. α_s) has to also be increased in order to get better accuracy. Figure 7.7 shows a comparison of the error convergence obtain with and without the use of FP, for regular nodal distribution. Figure 7.8 shows similar results for irregular set of nodes. In both cases we control the number of nodes in the support domain. Once again, MLS and WEN are rather unstable for irregular nodal distribution. Paradoxically, the effect seems to be stronger when the density of nodes increases.

7.3.2 Modified shape functions with derivatives independent variables

Another method to enforce derivative boundary conditions in Meshfree strong-form methods is to consider the normal derivatives of the field function at the DB-nodes as independent variables in the function approximation. This is referred to the Hermite collocation method (HCM). Taking the RPIM formulation (see section 4.2.4 p.68) as the starting point, we add the normal derivatives

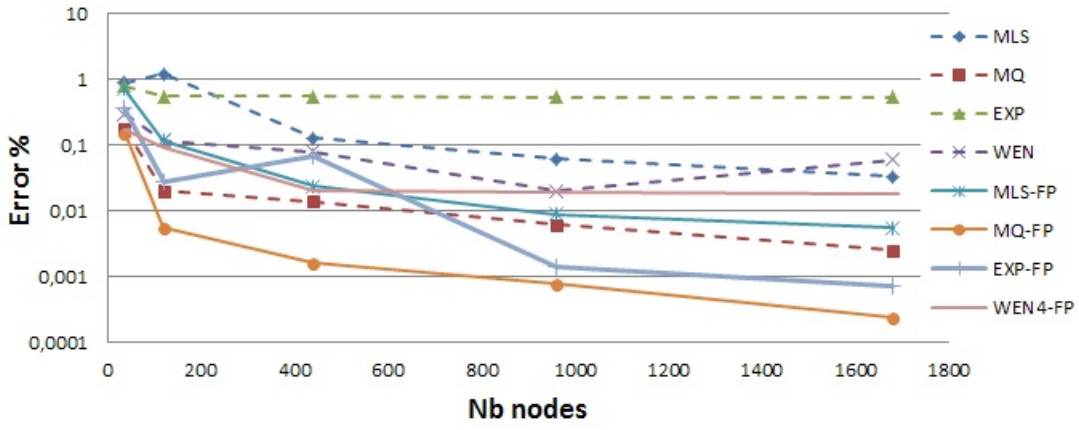


Figure 7.7: Convergence of collocation methods with direct and FP DBCs implementation methods for regular nodal distribution

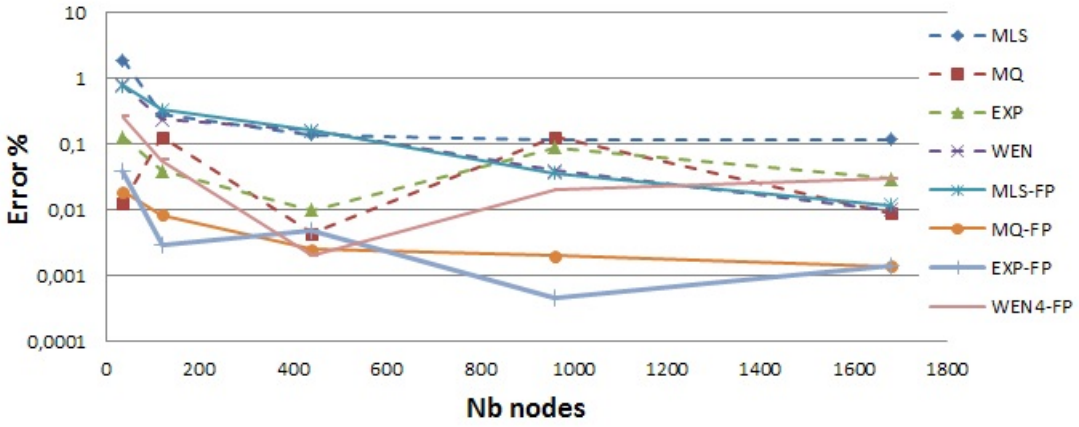


Figure 7.8: Convergence of collocation methods with direct and FP DBCs implementation methods for irregular nodal distribution

as extended degrees of freedom at the DB-nodes. We can write the new approximation including the normal derivatives at the DB-nodes

$$u^h = \sum_{i=1}^{n} R_i(\mathrm{x}) a_i + \sum_{j}^{m} p_j(\mathrm{x}) b_j + \sum_{k=1}^{n_{\mathrm{DB}}} \frac{\partial R_k^{DB}}{\partial \mathrm{n}} c_k \tag{7.24}$$

Taking the normal derivative of the field function at a DB-node and using the chain rule, we have

$$\frac{\partial R_k^{\mathrm{DB}}}{\partial \mathrm{n}} = n_{x_k} \frac{\partial R_k^{\mathrm{DB}}}{\partial \mathrm{x}} + n_{y_k} \frac{\partial R_k^{\mathrm{DB}}}{\partial \mathrm{y}} \tag{7.25}$$

n is the unit outward normal vector to the Derivative boundary and n_{x_i} and n_{x_i} are projections of the normal along x and y directions at the DB-node. a_i , b_j and c_k are 3 sets of coefficients to be determined. n, m and n_{DB} are the number of nodes in the local support domain, polynomial terms for augmentation and the number of DB-nodes within the local support domain respectively. We recap that for the RBFs

$$R_i(\mathrm{x}) = R(\|\mathrm{x} - \mathrm{x_i}\|) \tag{7.26}$$

$$R_k^{\mathrm{DB}}(\mathrm{x}) = R(\|\mathrm{x} - \mathrm{x_k^{DB}}\|) \tag{7.27}$$

We write equation 7.24 in the following matrix form

$$u^h(\mathrm{x}) = \mathrm{B^T a_0} \tag{7.28}$$

B^T being the vector of basis function

$$B^T = \left\{ \begin{array}{cccccccccc} R_1 & \cdots & R_n & \frac{\partial R_1^{\mathrm{DB}}}{\partial \mathrm{n}} & \cdots & \frac{\partial R_{n_{\mathrm{DB}}}^{\mathrm{DB}}}{\partial \mathrm{n}} & 1 & x & y & \cdots & p_m(\mathrm{x}) \end{array} \right\} \tag{7.29}$$

and the vector a

$$a_0^T = \left\{ \begin{array}{cccccccccc} a_1 & \cdots & a_n & b_1 & \cdots & b_{n_{\mathrm{DB}}} & c_1 & \cdots & c_m \end{array} \right\} \tag{7.30}$$

Similarly to the standard RPIM interpolation scheme, determining the coefficient can done by enforcing the interpolation (equation 7.24) to pass by each node within the support domain

$$u_l = u_{\mathrm{x_l}}^h = \left[\sum_{i=1}^{n} R_i(\mathrm{x_l}) a_i + \sum_{j=1}^{n_{\mathrm{DB}}} \frac{\partial R_j^{DB}}{\partial \mathrm{n}} b_j + \sum_{j}^{m} p_k(\mathrm{x_l}) c_k \right]_{l=1,\ldots,n_{\mathrm{DB}}} \tag{7.31}$$

At the DB-nodes we have

$$\frac{\partial u_l^{\mathrm{DB}}}{\partial \mathrm{n}} = \left[\sum_{i=1}^{n} \frac{\partial R_i(\mathrm{x_l^{DB}})}{\partial \mathrm{n}} a_i + \sum_{j=1}^{n_{\mathrm{DB}}} \frac{\partial^2 R_j^{DB}(\mathrm{x_l^{DB}})}{\partial \mathrm{n}} b_j + \sum_{k}^{m} \frac{\partial p_k(\mathrm{x_l^{DB}})}{\partial \mathrm{n}} c_k \right]_{l=1,\ldots,n} \tag{7.32}$$

In order to obtain a unique solutions for the sets of coefficient's equations, the following additional constrains are used [162]

$$\sum_{i=1}^{n} p_k a_i + \sum_{j=1}^{n_{DB}} p_k b_j = 0; \quad \forall k = 1, \ldots, m. \tag{7.33}$$

Using equation 7.31,7.32 and 7.33, one can now write the matrix form

$$U = \left\{ \begin{array}{c} U_s \\ \frac{\partial U_{DB}}{\partial \mathrm{n}} \\ 0 \end{array} \right\} = \left(\begin{array}{ccc} R & R_{DB} & P_m \\ R_{DB}^T & R_{DDB} & P_{DB} \\ P_m^T & P_{DB}^T & 0 \end{array} \right) \left\{ \begin{array}{c} a \\ b \\ c \end{array} \right\} = G a_0 \tag{7.34}$$

Where G is a generalized form of the moment matrix presented earlier in section 4.2.4 p.68. P_m^T is the regular polynomial moment matrix. Obtaining the coefficient a_0 necessitates the inversion of the G matrix. G is usually symmetric and invertible.

$$a_0 = G^{-1} U_s \tag{7.35}$$

Replacing this into equation 7.28 gives

$$u^h(\mathrm{x}) = \mathrm{B^T a_0} = \mathrm{B^T G^{-1} U_s} = \mathrm{\Phi^T U_s} \tag{7.36}$$

We obtain the SF vector $\mathrm{\Phi^T} = \mathrm{B^T a_0} = \mathrm{B^T G^{-1}}$ such

$$\Phi^T = \left[\begin{array}{ccccccccccc} \phi_1 & \cdots & \phi_i & \cdots \phi_n & \phi_1^H & \cdots & \phi_{n_{\mathrm{DB}}}^H & \phi_1^p & \cdots & \phi_m^p \end{array} \right]_{(n+n_{\mathrm{DB}}+m)\times l} \tag{7.37}$$

We can use these shape functions to retrieve the approximated field function

$$u^h(\mathrm{x}) = \sum_{i=1}^{n} \phi_i u_i + \sum_{j=1}^{n_{\mathrm{DB}}} \phi_j^H \frac{\partial u_j^{\mathrm{DB}}}{\partial \mathrm{n}} \tag{7.38}$$

And similarly for its derivatives

$$\frac{\partial u^h(\mathrm{x})}{\partial x} = \sum_{i=1}^{n} \frac{\partial \phi_i}{\partial x} u_i + \sum_{j=1}^{n_{\mathrm{DB}}} \frac{\partial \phi_j^H}{\partial x} \frac{\partial u_j^{\mathrm{DB}}}{\partial \mathrm{n}} \tag{7.39}$$

The existence of G^{-1} ensure no singularity in computing the HRPIM shape functions. HRPIM has been reported to be effective in dealing with arbitrary nodal distributions [159]. In calculating the SF, direct inversion of the Moment matrix can be avoided.

$$\Phi(\mathrm{x})G = B^T G^{-1} G \tag{7.40}$$

thus

$$G^T \Phi(\mathrm{x}) = B \tag{7.41}$$

Due to the non-singularity of G, this operation can be done using a standard linear solver.
Unlike for the EFG where the compatibility of the shape function is required over the whole domain, in collocation methods, it is only needed over support domain. A simpler form of MLS such as the weight least square (WLS) method can also be used to build SFs. It can be also derived in Hermite interpolation form. Similarly to MLS, they do not possess the delta function property, which is not a problem for collocation method where we use direct interpolation method for essential boundaries conditions. The HWLS procedure is fully described in [162].

7.4 benchmark tests

The following tests have been implemented in Matlab code and C++.

7.4.1 1D truss benchmark test

We consider the 1D truss bar problem governed by the equation

$$EA\frac{d^2u}{dx^2} + b(x) = 0 \tag{7.42}$$

with the essential boundary condition $u(0) = 0$ and the DBC or force boundary condition

$$\frac{du(L)}{dx} = -k\cos(k\pi) \tag{7.43}$$

With E the young modulus and A the cross section of the beam. u represents the solution for the axial displacement. The body force $b(x)$ is in the x direction. For simplicity, in the following test, we use E=1.0 and $A = 1.mm^2$. The length of the beam L is taken 1mm. The following source term is considered

$$b(x) = -(k\pi)^2 \sin(k\pi x) \tag{7.44}$$

The analytical solution of this problem is given by

$$u^e = -\sin(k\pi x) \tag{7.45}$$

In order to compare the effect of Hermite interpolation for the construction of SFs, we compare the results obtained from problem with the following DC imposed at x=L

$$u(L) = -\sin(k\pi) \tag{7.46}$$

Both problems have the same analytical solution given by equation 7.45. In the following we consider $k = 2.4$.
We first study the h convergence of collocation methods with MLS and WLS SFs, WEN4 RPIM, EXP RPIM and MQ RPIM. For now we consider the problem with only Dirichlet condition (DC). For WLS and MLS, the cubic spline weight function is used. Through this first test, no difference with the use of other weight functions such as quartic, new quartic or exponential has been found. $\alpha_s = 1.5$ with an average of 3 nodes per support domains. We consider the regular distribution of 5,20,30,40 nodes. Figure 7.9 shows the results obtained. As previously observed for data fitting

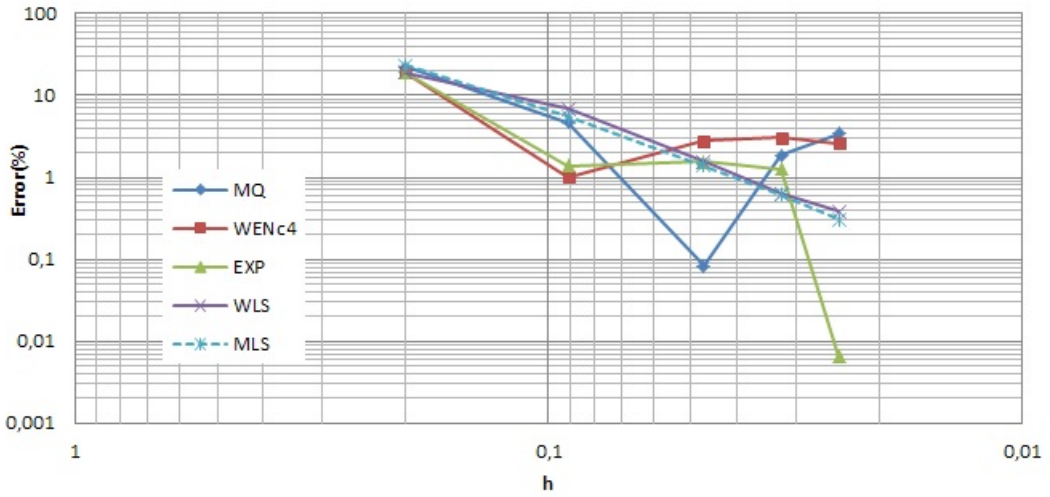


Figure 7.9: Convergence of RPIM, MLS and WLS for $\alpha_s = 1.5$.

and applications through weak formulation based methods, MLS works better than RPIM for a small number of nodes within the support domain. Here, the non-convergent behaviour of the RPIM can be explained by the overly small number of nodes present in the support domain. In order to verify the preliminary results from chapter 4.3, we now study the impact of different α_s on the accuracy (error $\%$). A 31 regular nodes distribution is used for this test. Table 7.2 gives the average number of nodes within a support domain for each values of α_s For MLS and WLS,

α_s	1,5	2	2,5	3	3,5	4	4,5	5	6	7	8	9	10	11
Nb nodes	3	4	5	6	6	7	8	9	10	12	13	15	16	17

Table 7.2: Average number of nodes in support domain for different values of α_s

second order basis is used to compare the results for this two methods. As shown on figure 7.10, both methods give similar results. Increasing the parameter α_s brings more nodes in the support domain. However it tends to decrease the accuracy of both MLS and WLS approximation when

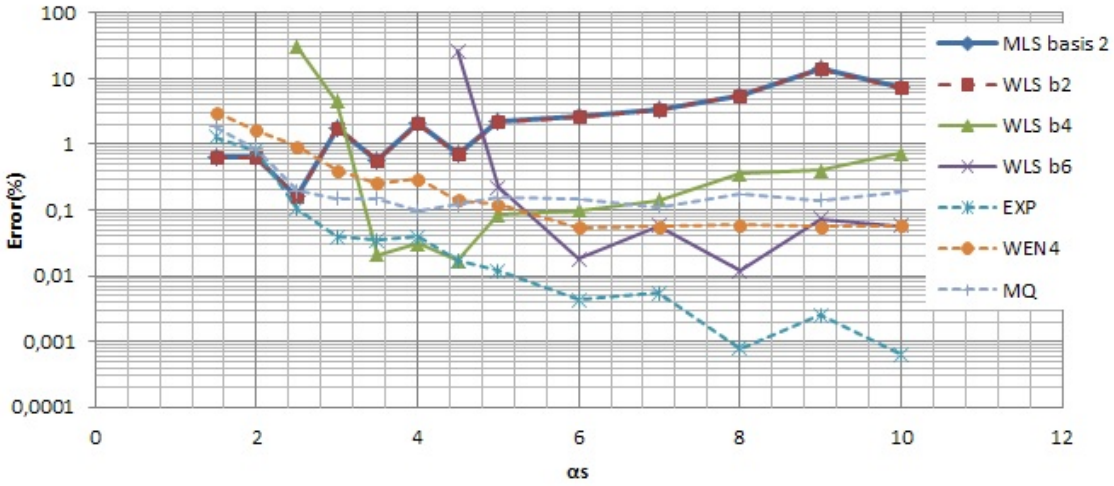


Figure 7.10: Impact of α_s on the accuracy of MLS and WLS and RPIMs collocation methods

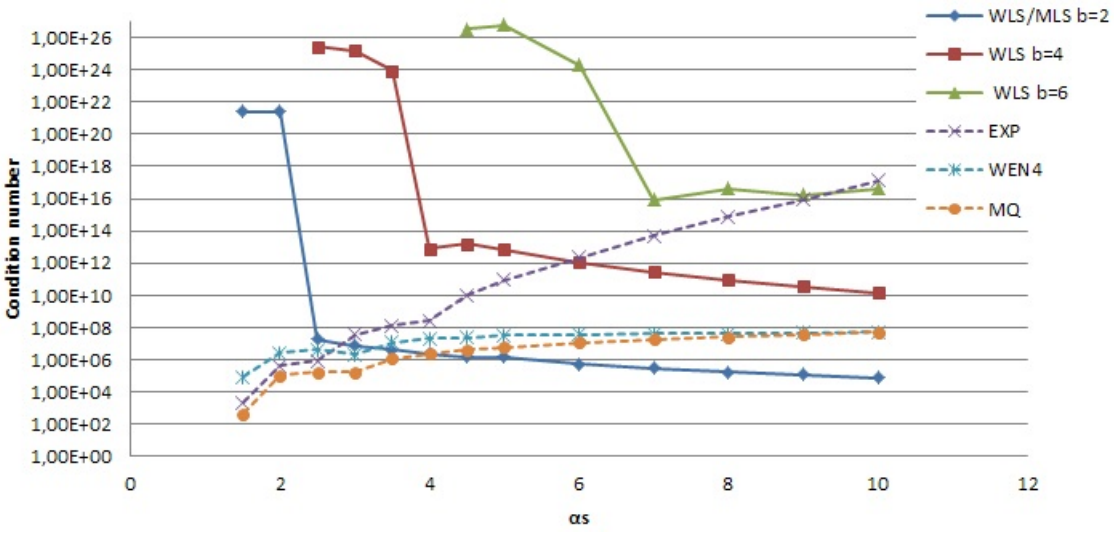


Figure 7.11: Impact of α_s on the condition number of the moment matrix of MLS,WLS and RPIMs SFs

too many nodes are involved. As mentioned in section 4.3.2, MLS works well for a relatively small number of nodes within the support domain Ω_s. Figure 7.10 shows that in order to keep the accuracy when more nodes are introduced, the order of the basis needs to be increased. It appears that there is a limit where increasing the order of the basis as well as the support domain size has a negative impact on the condition number of the Moment matrix. When the number of nodes is too low for the order of the basis the matrix can be close to singular if the precision is limited. In our case, the strategy of controling the number of nodes limit the computation charges so it should be preferred. In practice, increasing the order of the basis should be the solution to avoid the nodes elimination algorithm for instance.

Similar tests are carried out for the RPIM SFs. As expected, RPIM collocation provides better accuracy as more nodes are introduced within the support domain. As shown on figure 7.10, the EXP SFs gives the best accuracy then MQ or WEN RBFs. However condition number of the Moment matrix (figure 7.11) is much bigger but still remains manageable. This test illustrates the sensitivity of the methods to the number of nodes within the support domain and translates

to a tendency in practice to be rather sensitive to nodes irregularity when no treatment is done. Optimization should be imposed by controlling the nodes within the support.
We now use an irregular set of nodes with irregular factor $Ifc = 0.6$. Based on the previous tests, we control the number of nodes within the support domain. For MLS and WLS number of nodes are 5-6 and for RPIMs between 12-14. Figure 7.12 show the results obtained. Through

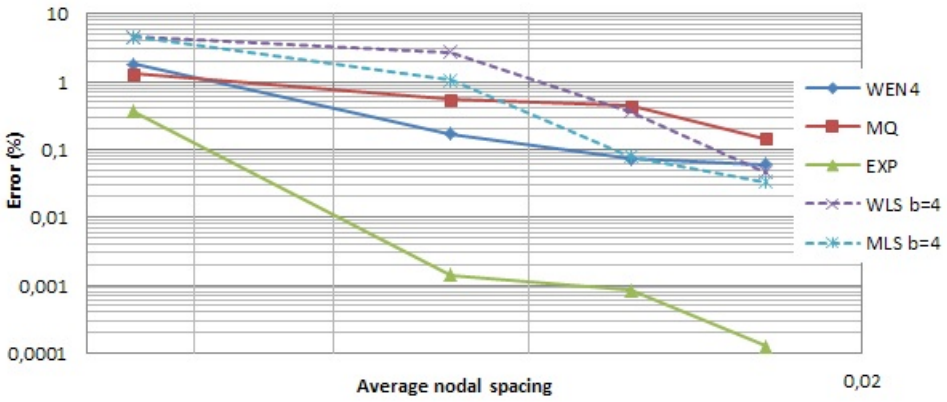


Figure 7.12: Convergence of RPIM, MLS and WLS collocation methods for irregular distribution of nodes.

this test for irregular distribution of nodes, all the methods provide acceptable convergence and accuracy provided that control of support domain and parameters are done. The RBFs seems to provide slightly more accurate solution as well as less sensitivity to the irregularity of the nodal distribution.
A noticeable feature about WLS and MLS is that the basis order should be higher for collocation methods. For irregular set of nodes when second order derivative is needed in the strong form, 3^rd or 4^th order basis provides optimal results. By controlling the number of nodes we can decrease the basis if too few nodes are present in the support domain to avoid encountering badly conditioned Moment matrix as shown in the previous test (figure 7.11).
The condition number of the moment matrix benefits also from controling the number of nodes in support domain (figure 7.13). Because compatibility over the whole domain is not needed for

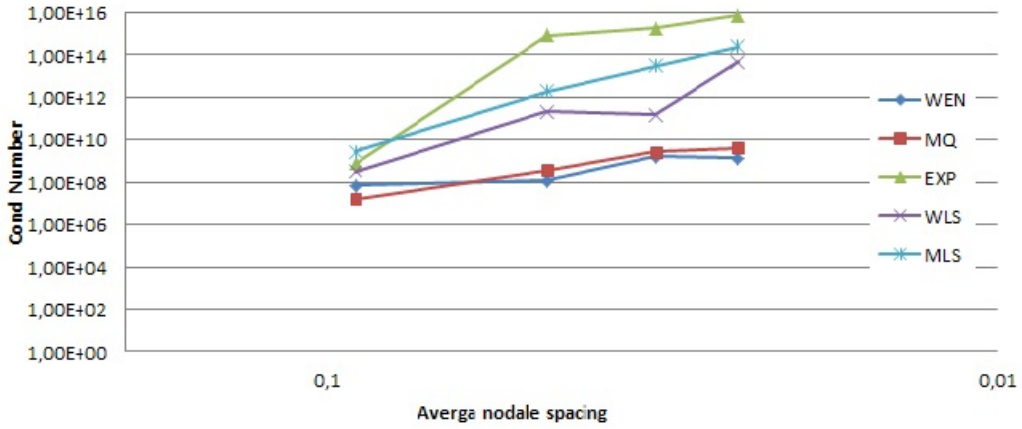


Figure 7.13: Convergence of RPIM, MLS and WLS collocation methods for irregular distribution of nodes.

collocation methods, and the fact that WLS requires less computation while providing similar

accuracy, WLS can be selected for the application to reduce the computation of SFs.
We now compare some of the methods that deal with DBC correction, e.g. Direct implementation (DBC), FP, and HCM type SFs. We study the effect for the col-RPIM -MQ for regular (figure 7.14) and irregular nodal distributions (figure 7.15). MQ-DC refers to the solution of the equivalent problem obtained with only Dirichlet boundary conditions. MQ-DBC is the solution obtained by a direct replacement of the i^th line of the system matrix by the derivative condition equation at the i^th nodes. This means no particular treatment is operated.

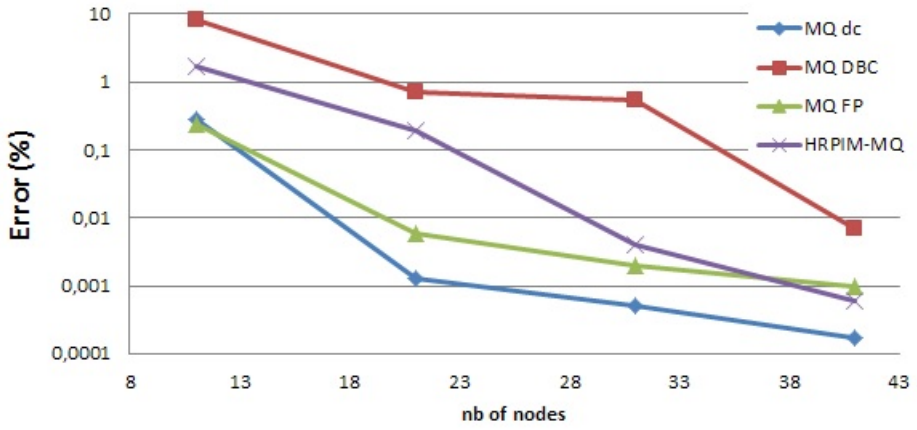


Figure 7.14: Convergence of col-RPIM MQ for different DBC implementation methods. Regular nodal distribution, MLS and WLS collocation methods for irregular distribution of nodes.

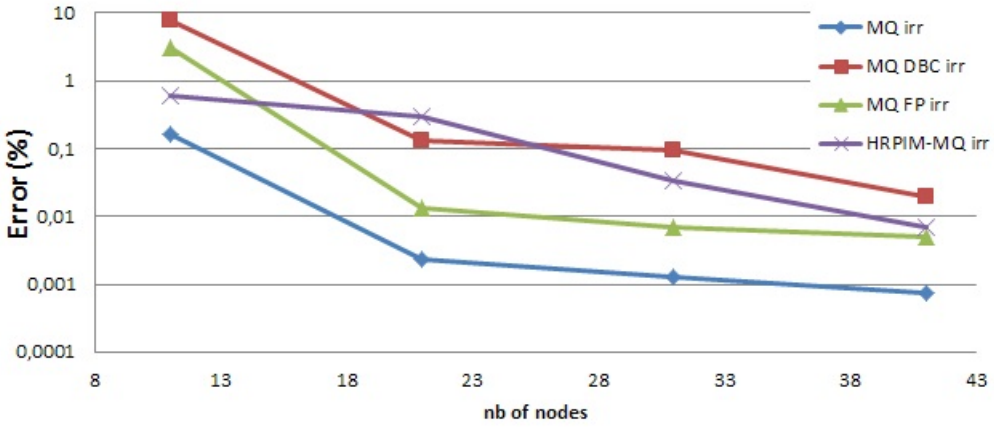


Figure 7.15: Convergence of col-RPIM MQ for different DBC implementation methods and regular nodal distribution

We can observe that both FP and HCM improve the error by enforcing DBC more accurately. Similar results are obtained for HRPIM-EXP. Figure 7.16 shows the results for regular nodal distribution and figure 7.17 for irregular distribution. The FP methods give good results considering the ease of implementation and small computational cost added by the methods. Both correction methods works well for this 1D dimensional problem. However these results need to be confirmed for higher order problems. Overall, RPIM techniques are quite stable for irregular nodal distribution and can provide even better result for irregular sets.
On figures 7.14, 7.15, 7.16 and 7.17, it can be observed that as the nodal density increases, the accuracy of the methods for direct implementation of DBC improves as well. This is another technique that can be used to improve the enforcement of DBC by adding a higher nodal density

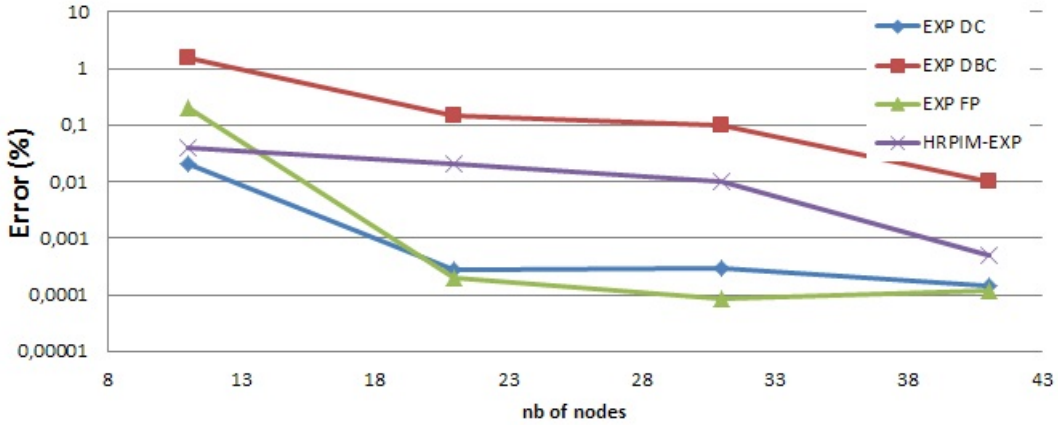


Figure 7.16: Convergence of col-RPIM EXP for different DBC implementation methods and regular nodal distribution

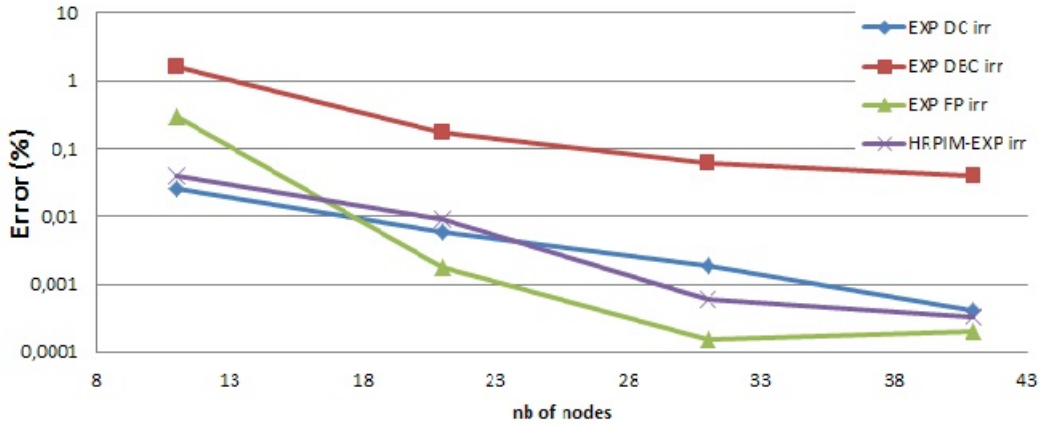


Figure 7.17: Convergence of col-RPIM EXP for different DBC implementation methods and Irregular nodal distribution

close to these boundaries. In case of regular nodal distribution it also corresponds to a technique that relies on adding regular nodal layers close to the boundary. This why we can observe a good increase in accuracy for the DBC methods, for regular nodal distribution on figure 7.14 and 7.16. We also tested the methods for WLS SFs. Figure 7.18 and 7.19 shows the convergence of error for regular and irregular nodal distributions respectively.

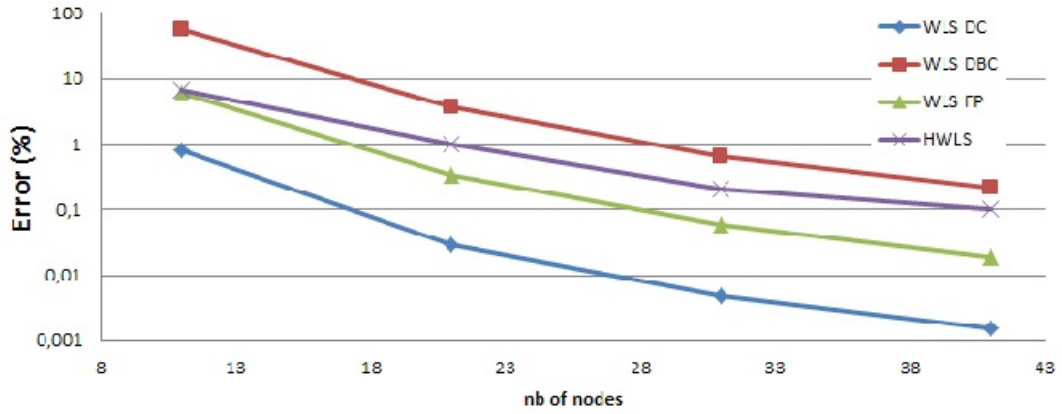


Figure 7.18: Convergence of col-WLS for different DBC implementation methods and regular nodal distribution

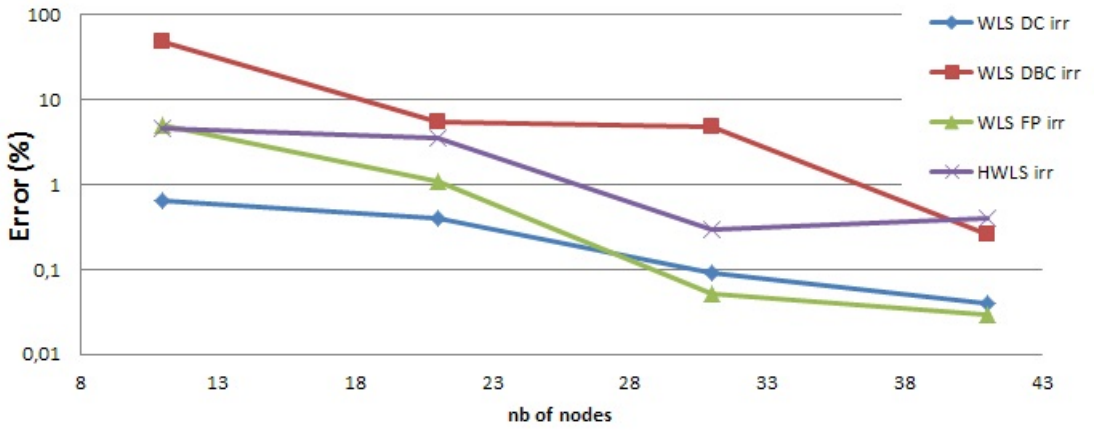


Figure 7.19: Convergence of col-WLS for different DBC implementation methods and Irregular nodal distribution

Both WLS and HWLS have been found to be rather unstable for an irregular set of nodes. This phenomenon is made worse when the number of nodes within the support domain is not controlled. The main issue with HCM and adding derivatives at the DBC nodes in the basis is that it requires managing additional lists of data. Detection of boundary nodes within a domain of influence is added to the regular task previously described in the process of building shape functions. If the order of the derivative in the strong form is N, there is also the need in the case of HRPIM to calculate the normal derivative of order n+1 to compute normal moment derivatives (equation 7.32 p.179). Hence the building of these SFs is more complex and more costly.

7.4.2 Two dimensional Laplace problem

We consider the 2D heat conduction represented by the potential problem in section 6.1, page 131.We first compared the convergence obtained for different Sfs with collocation (CO) for both regular and irregular set of nodes. Figure 7.20 illustrates how an irregular node distribution impacts the accuracy of the methods. Through this test, the Co-MLS shows less accurate results for both regular and irregular sets of nodes. It has also been found to be unstable for irregular sets. Small changes in the nodal distribution affects the accuracy. Node control tends to improves the accuracy

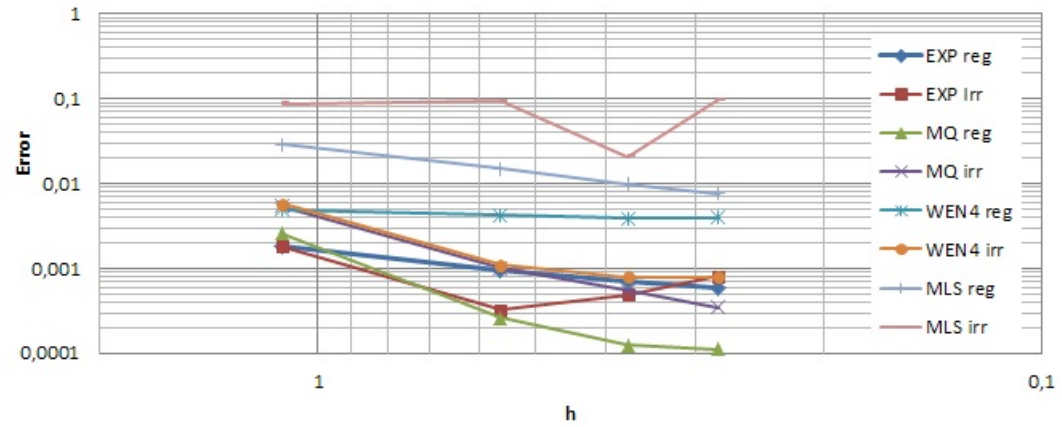


Figure 7.20: Comparison of convergence of Collocations methods for regular and irregular distributions of nodes.

but the presence of the DBC degrades the behaviour of the methods. RPIM SFs on the other hand are much more stable. Figure 7.21 shows the improvement when using fictitious point methods for dealing with the DBC imposition. As depicted by figure 7.21, FP method greatly improves the

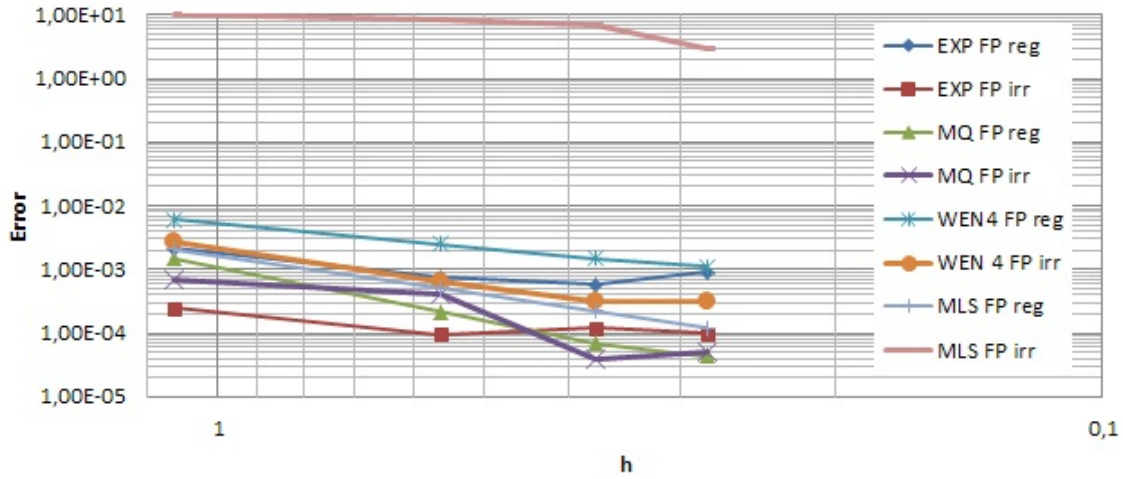


Figure 7.21: Comparison of convergence of Collocations methods using Fictitious points for regular and irregular distributions of nodes.

accuracy by imposing the DBC condition more accurately than the directed collocation method. However no benefit was found on MLS for irregular set of nodes.

We now compare the collocation methods results with EFG using both MLS and RPIM SFs as well as FEM using QUAD element. We developed our own FEM Matlab code in order to compare the different methods. Collocation methods use Fictitious point (FP) to deal with the DBC. Figure 7.22 shows a comparison of the convergence of collocation methods with EFG and FEM. h denotes the average refinement of nodal spacing. Figure 7.23 compares the computational time for the different methods.

Figure shows that collocation methods are faster than the weak formulation based methods. This is due to the absence of integration. However the RPIM SFs were tuned for accuracy. Optimised support domains were used for better accuracy, hence the Meshfree SFs were more costly to compute. This is mainly due to the number of nodes involved, resulting in a bigger Moment matrix. This causes the collocation methods to be slower than the standard QUAD FEM. A trade-off has

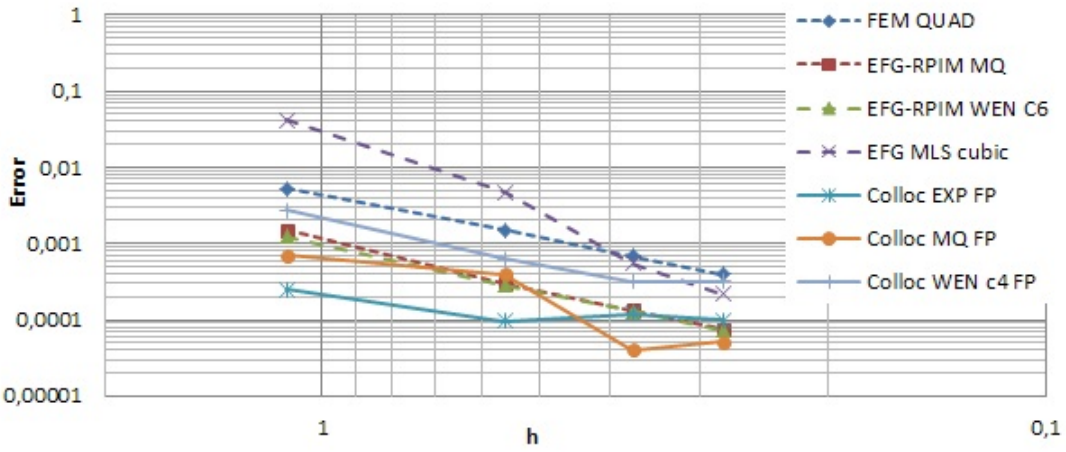


Figure 7.22: Comparison of the convergence of error for different collocation methods and FEM

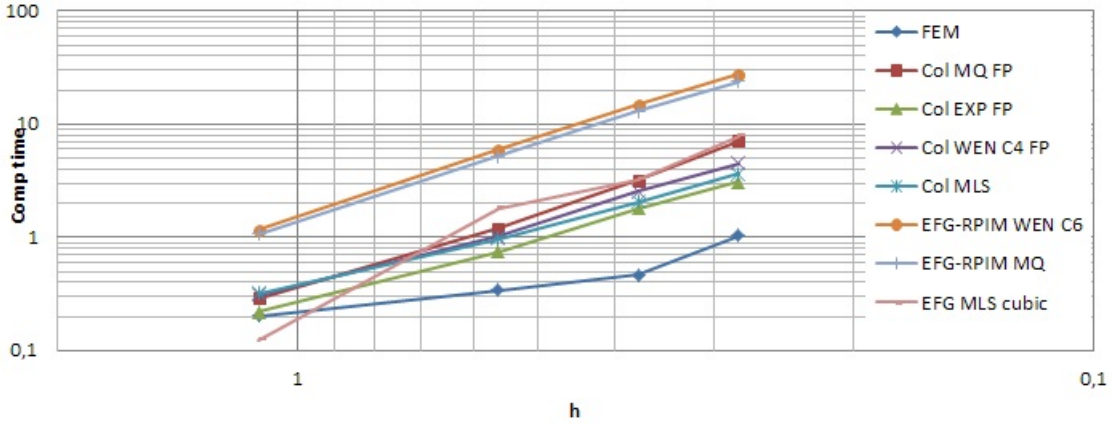


Figure 7.23: Comparison of computational time for collocation method and FEM

to be made between accuracy and speed. We now set the parameters in a way to reduce the computational time of the Meshfree SFs by limiting the support domain. Figure 7.24 shows the results for the Collocation RPIM MQ-FP.

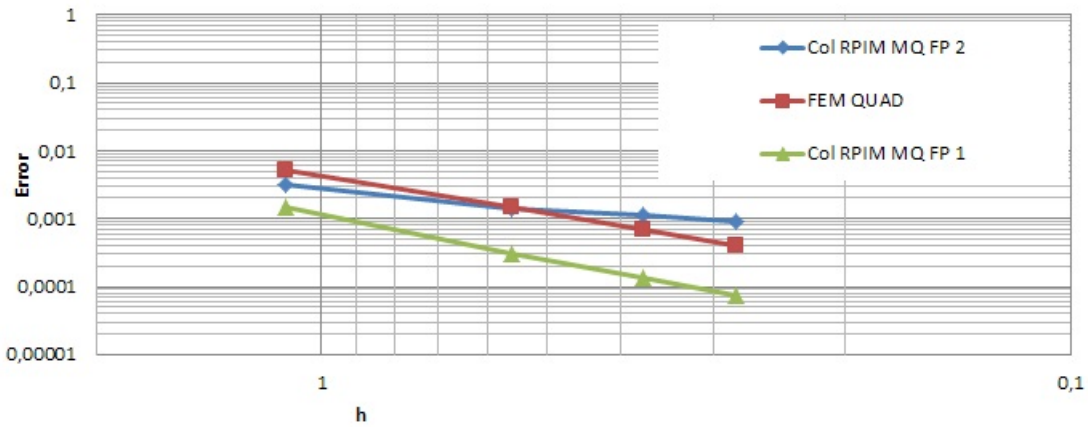


Figure 7.24: Comparison of the convergence of col-RPIM PF with optimised paramter for accuracy (FP1) and computational time (FP2)

The Col RPIM MQ FP 1 represents the collocation method using RPIM MQ with parameters optimised for accuracy whereas Col RPIM MQ FP 2 stands for the fast computation version.

As shown in Figure 7.24, reducing the number of nodes within the support domain reduces the accuracy. However figure 7.25 demonstrates a huge improvement in computational time with the methods being now slightly faster than the FEM one.

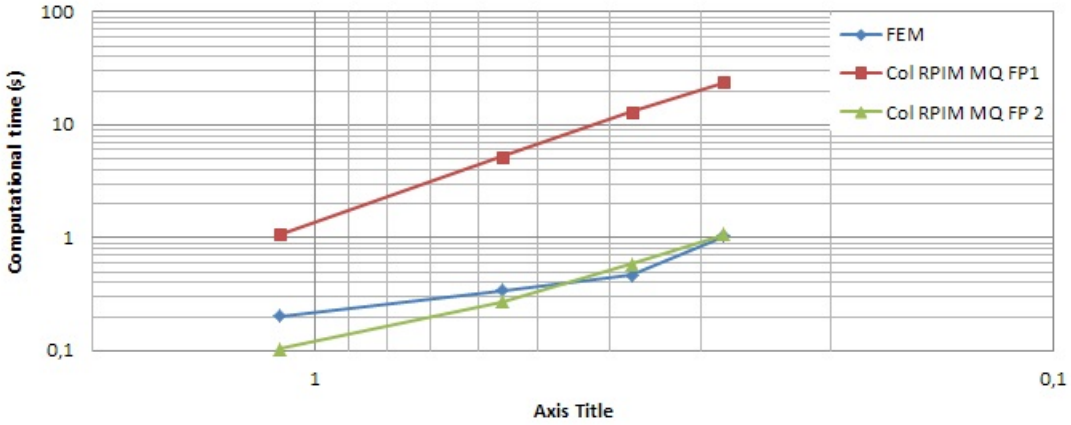


Figure 7.25: Comparison of computational time for FEM and collocation MQ for fast computation.

This are good results, as here the FEM QUAD algorithm does not involve time consuming mesh management. The FE bilinear SFs over isoparametric elements are simple and fast to compute. They tend to produce very good results for Elliptic problem such as the Laplacian. Hence, the collocation demonstrates in this case, good potential in treating such problems accurately and quickly. In practical problems with much more DOFs and moving boundaries the collocation would have the advantage of not needing mesh compared to the FE.

7.4.3 Transient heat conduction

We consider the Heat transfer problem governed by equation 6.65 page 152. We use the standard Crank-Nicholson(CN) scheme to perform the time derivation. Using the collocation procedure on equation 6.65 gives the following recurrence system

$$\left(C - \frac{\Delta t}{2}K\right)U^t = \left((C + \frac{\Delta t}{2}K\right)U^{t-1} \tag{7.47}$$

With U^t and U^{t-1} being the nodal values vector at $t-1$ and t. The matrix C and K are the stiffness matrix and the damping matrix obtained using collocation procedure where the nodal matrix at x_I is given by

$$K_I = \sum_{j=1}^{n} \frac{\partial^2 \phi_j}{\partial x^2}\frac{\partial^2 \phi_j}{\partial y^2}; \quad C_I = \sum_{j=1} \phi_j \tag{7.48}$$

With n being the number of nodes within the support domain of the collocation point x_I. The boundary conditions are applied using standard collocation procedure. Figure 7.26 shows the temperature solution at $t = 0.12s$ obtained with collocation MQ with 441 regular nodes distribution with a time step of 0.001s. The method shows very good convergence with an error of $0,001\%$. Figure 7.27 displays the variation of error through time. The average error over all sampling times is 0.06%. We use the parameter defined during our study for a steady state conduction problem in section 7.4.2.

The time period where the solution is less accurate corresponds to when the temperature field is

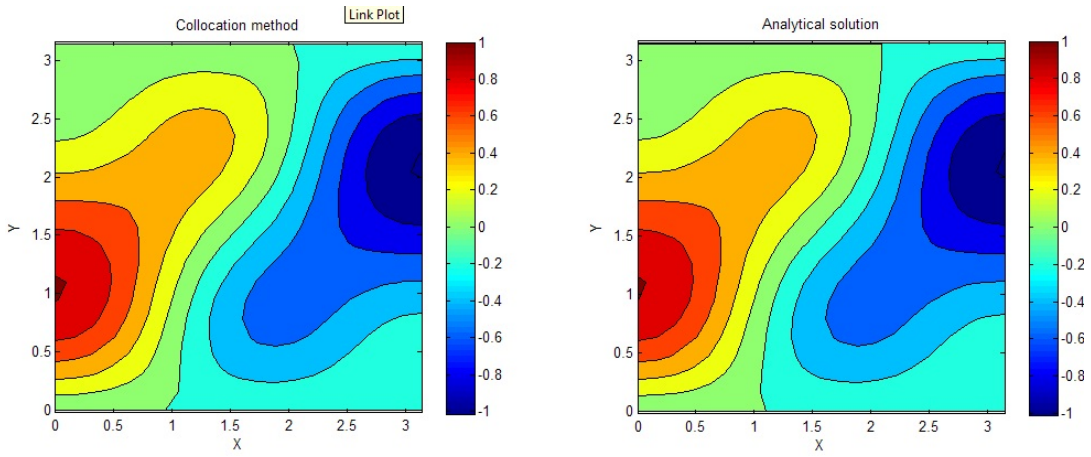


Figure 7.26: Temperature field at 0.12 s obtained with Col RPIM-MQ. 441 regular nodes distribution. Error = 0.001%

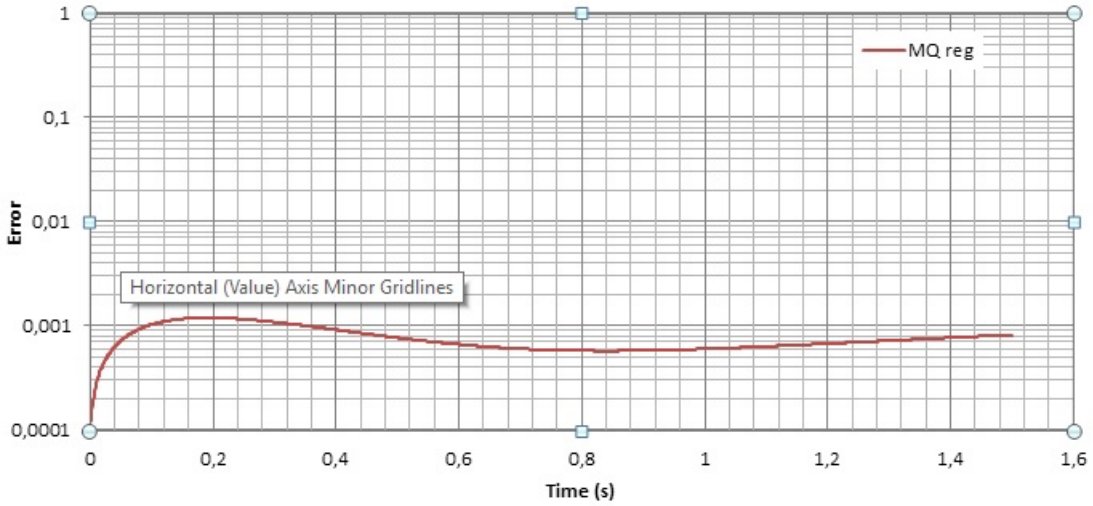


Figure 7.27: Variation of error through time with Col RPIM-MQ. 441 regular nodes distribution. Average Error=0.06%

rapidly changing. Once the temperature field is more stable the solution provides accuracy of a good order similar to the one observed for steady state problems. We study the influence of the Δt on the accuracy. Figure 7.28 shows that $\Delta t = 0.01s$ is sufficient for reaching good accuracy. Finer time steps do not noticeably improve the accuracy. Similar results are obtained using other Sfs. We now study the convergence of the solution for different nodes density. Regular distribution of 6×6, 11×11, 21×21 , 31×31 and 41×41 nodes are used. Figure 7.29 shows the results, and demonstrates that the accuracy increases along the node density.
For the RPIM Sfs no benefits were found by adding polynomial basis.

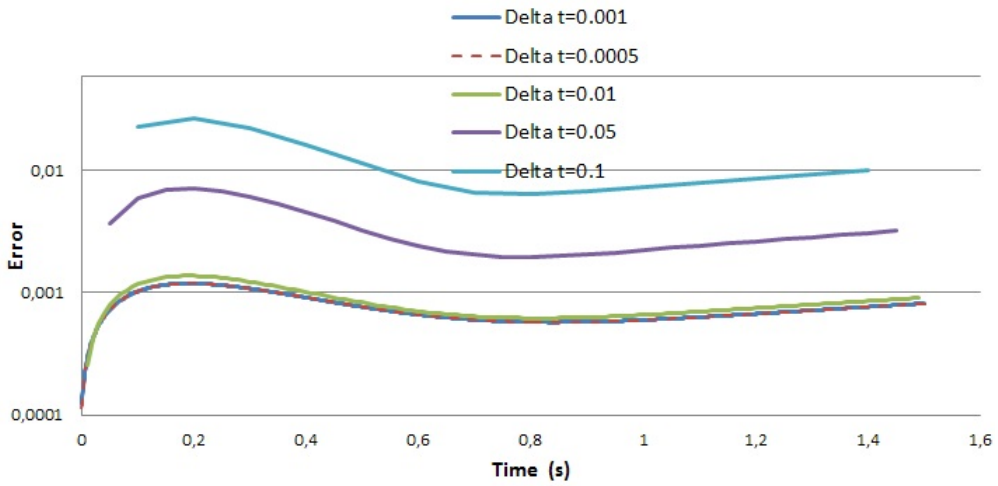


Figure 7.28: Influence of δt on the error. Col RPIM-MQ. 441 regular nodes distribution

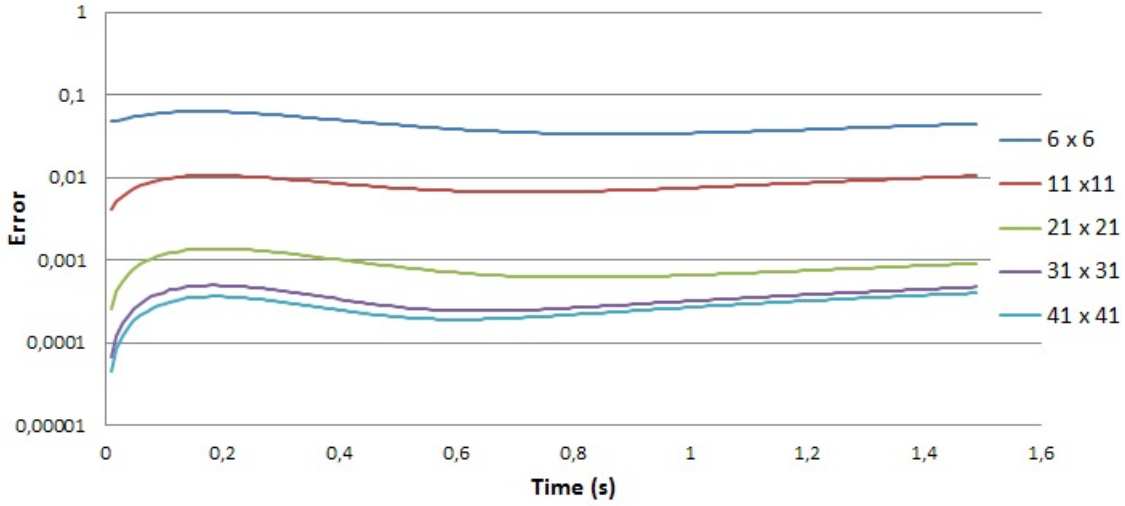


Figure 7.29: Convergence of the Error. Col RPIM-MQ

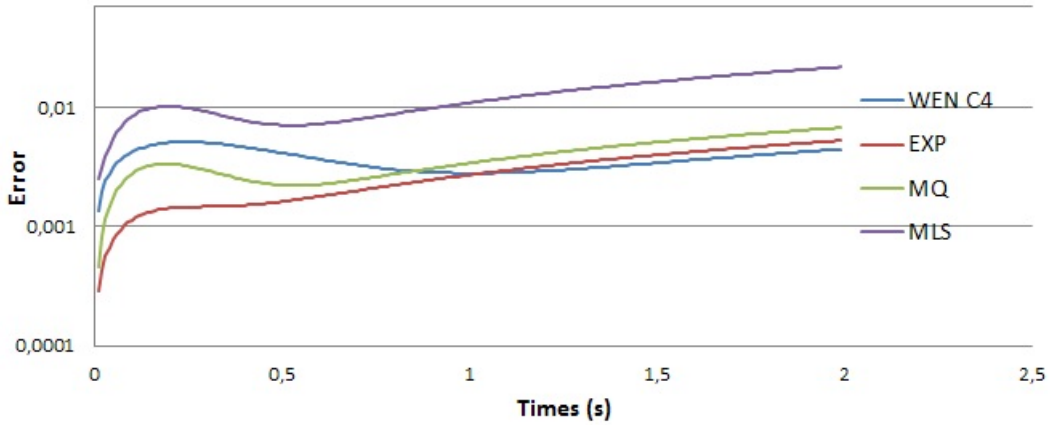


Figure 7.30: Comparison of accuracy for different Sfs. 21×21 regular distribution of nodes

However it has been found that in order for the RPIM to produce acceptable accuracy and stability, a controlled number of nodes $n \geq 20 - 24$ for RPIM-EXP and MQ works well. We also study the effect on accuracy of the imposition of DBC using FP technique. Figure 7.31 shows the results obtain for an irregular set of 441 nodes.

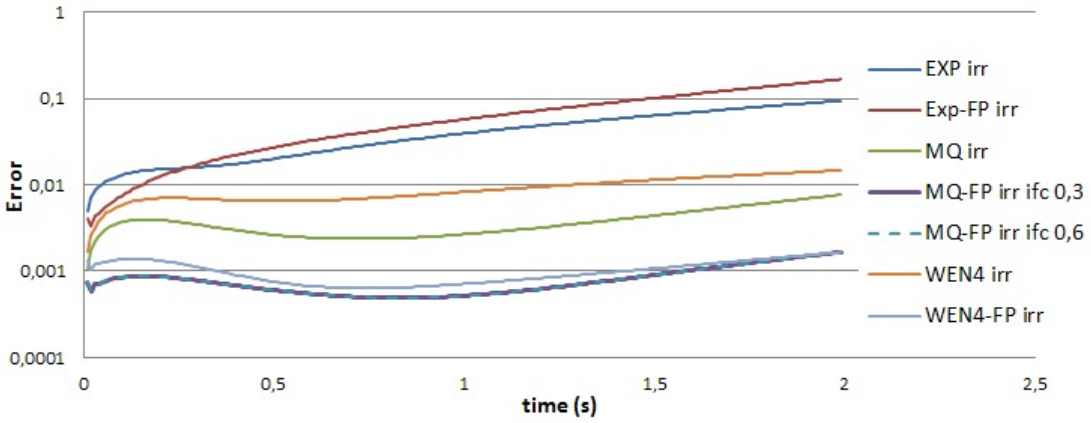


Figure 7.31: Comparison of accuracy for different Sfs with fictitious points. 21×21 irregular distribution of nodes

For Col-RPIM-MQ, using FP techniques to impose DBC for both regular and irregular distribution of nodes improves the accuracy significantly. This also results in a greater stability and robustness of the method overall. More nodes ($\approx 30 - 32$) in the support domain seems to provide better accuracy in this case. Slightly less benefits is observed for the CSRBF-WEN C4 functions.
In the case of col-RPIM-EXP a bigger parameter $\alpha_c = 0.3$ works better for irregular distribution of nodes. Using fictitious points for DBC does not significantly improve the accuracy in the case of irregular node distribution. In the case of MLS, no improvement has been found on the accuracy and overall the method does not works well for irregular nodal distribution.

7.4.4 Non-linear time dependent 2D heat conduction problem

We consider the problem from section 6.5 where heat capacities and thermal conductivities depend on temperature. The PDE for this problem is given by equation 6.70 154. Using the similar Kirchhoff transformation (equation 6.75), the problem is transformed into

$$\gamma(u)\frac{\partial u}{\partial t} - \nabla^2 u - Q(x,t) = 0 \quad \forall x \in \Omega \tag{7.49}$$

with$\gamma = \rho C/k$ called the reciprocal of thermal diffusivity.

$$u(x,t) = \overline{u}(x,t), \quad \forall x \in \Gamma_1 \tag{7.50}$$

$$\frac{\partial u}{\partial n} = \overline{q}(x,t) \cdot n, \quad \forall x \in \Gamma_2 \tag{7.51}$$

and

$$u(x,0) = \overline{u}_0(x), \quad \forall x \in \Omega \tag{7.52}$$

We first use collocation discretisation for the $i = 1...N$ nodes and the explicit forward difference (see section 6.55 p. 151) approximation for the time derivative of equation 7.49. We obtain the following form

$$\theta\nabla^2 u_i^n - \gamma' \frac{u_i^n}{\Delta t} = -(1-\theta)\nabla^2 u_i^{n-1} - \gamma' \frac{u_i^{n-1}}{\Delta t} - Q(x,(n-1+\theta)\Delta t) \quad \forall x \in \Omega \tag{7.53}$$

With $\gamma' = \gamma(\theta u_i^n + (1-\theta)u_i^{n-1})$. The initial and boundary condition(IBC) equations 7.50-7.52

become

$$u_i^n = \overline{u}(x_i, n\Delta t), \quad \forall x_i \in \Gamma_1 \tag{7.54}$$

$$\frac{\partial u^n}{\partial n} = \overline{q}(x_i, n\Delta t) \cdot n, \quad \forall x_i \in \Gamma_2 \tag{7.55}$$

and

$$u_i^0 = \overline{u}_i^0, \quad \forall x \in \Omega \tag{7.56}$$

where u_i^n and u_i^{n-1} correspond to the nodal values of $u(x_i, n\Delta t)$ and $u(x_i, (n-1)\Delta t)$ respectively. Δt corresponds to the time interval between each time iteration. Using the MF approximation 7.11 on 7.53 and the IBC 7.54-7.56 gives

$$\theta\nabla^2 \sum_{j=1}^{N} \phi_j^i u_j^n - \frac{\gamma'}{\Delta t} \sum_{j=1}^{N} \phi_j^i u_j^n = -(1-\theta)\nabla^2 \sum_{j=1}^{N} \phi_j^i u_j^{n-1} - \frac{\gamma'}{\Delta t} \sum_{j=1}^{N} \phi_j^i u_j^{n-1} - Q(x, (n-1+\theta)\Delta t) \tag{7.57}$$

$$\phi_j^i u_j^n = \overline{u}(x_i, n\Delta t), \quad \forall x_i \in \Gamma_1 \tag{7.58}$$

$$\frac{\partial \phi_j^i u_j^n}{\partial n} = \overline{q}(x_i, n\Delta t) \cdot n, \quad \forall x_i \in \Gamma_2 \tag{7.59}$$

and

$$u_i^0 = \overline{u}_i^0, \quad \forall x_i \in \Omega \tag{7.60}$$

ϕ_j^i being the shape function for the j^{th} nodes in the support domain evaluated at the sampling point i.e collocation point x_i.
As described in section 6.5, we start the iteration by assuming that $u_j^n = u_j^{n-1}$ in order to compute the reciprocal thermal diffusivity γ'. We use $\theta = 0.5$ (crank-Nicholson) for the time derivative.
γ is interpolated using a piecewise linear function $\forall u \in [u_k; u_{k+1}[$

$$\gamma(u) = \frac{\gamma_{k+1} - \gamma_k}{u_{k+1} - u_k}(u - u_k) + \gamma_k. \tag{7.61}$$

With γ_k and u_k being the values obtained at the sample points for the material properties chart.
Figure 7.32 shows the error for both the temperature field and its gradient obtained with the col-RPIM MQ (red) , WEN-C4 (blue) and EXP (black) obtained for a regular set of 36 nodes distribution.
Similar observations to the previous problem can be made. Results similar to those obtained through the weak formulation methods.

7.4.5 2D elastostatic beam problem

We consider the same problem as in section 5.5, a Beam under a parabolic load at his free end. The governing equation of this problem is given by equation 5.1. Figure 7.33 shows the deflection of the consider beam obtained with the Col-MLS for a set of 121 irregular nodes. Using the constitutive equations and stress-displacement relation, the strong form equation5.1 for the plane

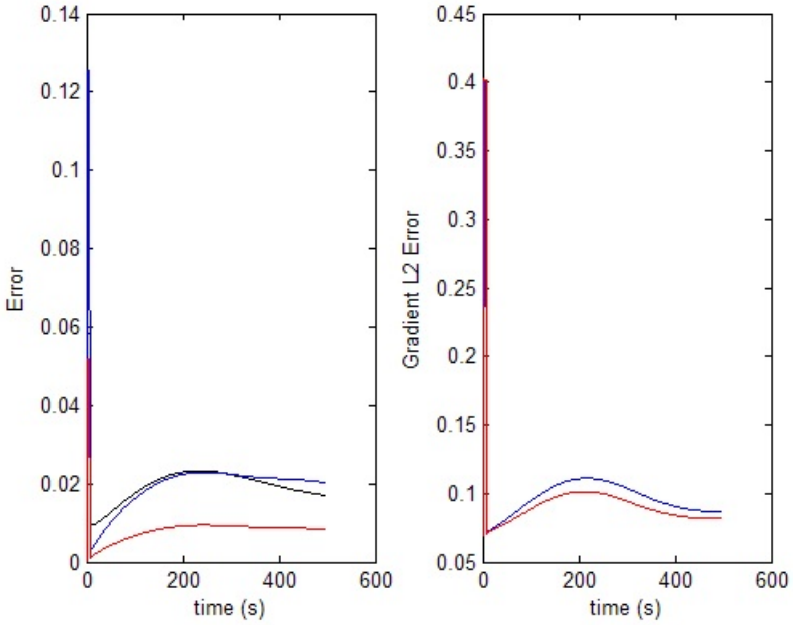


Figure 7.32: temperature field and gradient error

stress problem can be written in terms of displacements.

$$\frac{E}{1-\nu^2}\left(\frac{\partial^2 u}{\partial x^2}+\frac{1-\nu}{2}\frac{\partial^2 u}{\partial y^2}+\frac{1+\nu^2}{2}\frac{\partial^2 v}{\partial x \partial y}\right)+b_x=0$$
$$\frac{E}{1-\nu}\left(\frac{\partial^2 v}{\partial y^2}+\frac{1-\nu}{2}\frac{\partial^2 v}{\partial x^2}+\frac{1+\nu}{2}\frac{\partial^2 u}{\partial x \partial y}\right)+b_y=0 \tag{7.62}$$

With $b=\{b_x b_y\}^T$ the external body force vector. Using equations7.11 and 7.13 We have for a field node

$$\begin{Bmatrix} u \\ v \end{Bmatrix}=\sum_{i=1}^{n}\begin{bmatrix} \phi_i & 0 \\ 0 & \phi_i \end{bmatrix}\begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \tag{7.63}$$

$$\begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial x} \end{Bmatrix}=\sum_{i=1}^{n}\begin{bmatrix} \frac{\partial \phi_i}{\partial x} & 0 \\ 0 & \frac{\partial \phi_i}{\partial x} \end{bmatrix}; \tag{7.64}$$

With n is the number of nodes included in the support domain of the collocation point. u_i and v_i the nodal displacement at the i^th nodes in support domain. Higher order derivatives are obtained in similar manner. We can now write the following discretised system of equations for each field node.

$$\frac{E}{1-\nu^2}\sum_{i=1}^{n}\begin{bmatrix} \frac{\partial^2\phi_i}{\partial x^2}+\frac{1-\nu}{2}\frac{\partial^2\phi_i}{\partial y^2} & \frac{1+\nu}{2}\frac{\partial^2\phi_i}{\partial x\partial y} \\ \frac{1+\nu}{2}\frac{\partial^2\phi_i}{\partial x\partial y} & \frac{\partial^2\phi_i}{\partial y^2}+\frac{1-\nu}{2}\frac{\partial^2\phi_i}{\partial x^2} \end{bmatrix}\begin{Bmatrix} u_i \\ v_i \end{Bmatrix}=\begin{Bmatrix} b_x \\ b_y \end{Bmatrix} \tag{7.65}$$

This gives the system matrix for each collocation node x_I

$$K_I U=F_I \tag{7.66}$$

In this problem the DBCs correspond to prescribed traction forces $\bar{t}$. These stress boundary con-

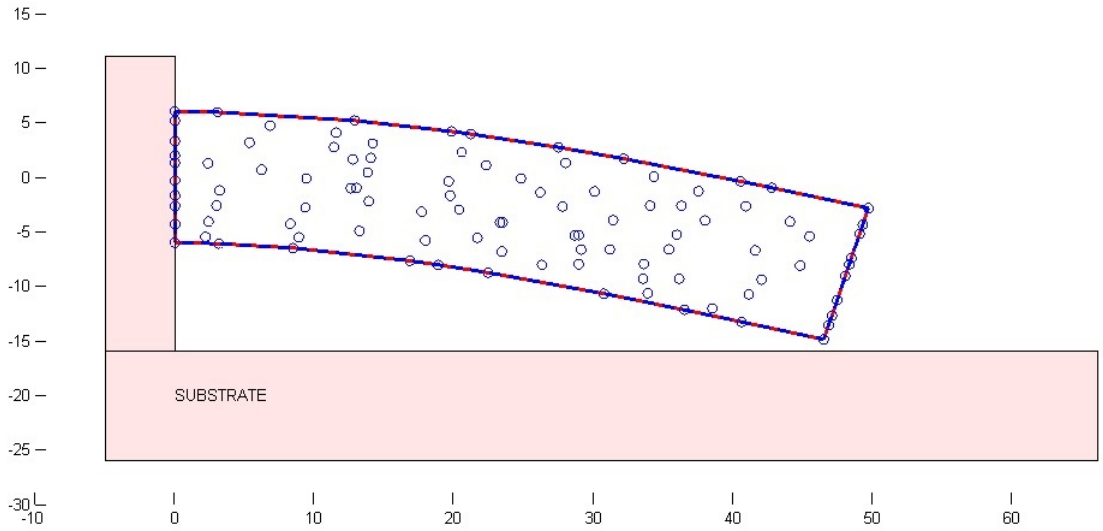


Figure 7.33: Beam deflection obtained with Col-RPIM-MQ with 121 irregular set of nodes. Error in displacement = $2.6 \times 10^{-4}\%$.

ditions on Γ_t can be written

$$t = \left\{ \begin{array}{c} \bar{t}_x \\ \bar{t}_y \end{array} \right\} = \left[\begin{array}{ccc} n_x & 0 & n_y \\ 0 & n_y & n_x \end{array} \right] \left\{ \begin{array}{c} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{array} \right\} \tag{7.67}$$

With **n**= $\{n_x, n_y\}$ the normal vector at the boundary Γ_t at the DB-node. $\bar{t}_x$ and $\bar{t}_y$ are the prescribed tractions force components in x and y coordinates. Using the stress-displacement relationship equation 5.8 gives the following strong form for the stress boundary condition at each DB-nodes.

$$\begin{aligned} \frac{E}{1-\nu^2}\left(n_x\left(\frac{\partial u}{\partial x}+\nu\frac{\partial v}{\partial y}\right)+n_y\frac{1-\nu}{2}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right) &= \bar{t}_x \\ \frac{E}{1-\nu^2}\left(n_y\left(\nu\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+n_y\frac{1-\nu}{2}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right) &= \bar{t}_y \end{aligned} \tag{7.68}$$

Using the collocation procedures similarly to obtain the system equation7.65, we have the following 2 equations at each DB-node.

$$\frac{E}{1-\nu^2}\sum_{i=1}^{n}\left[\begin{array}{cc} n_x\frac{\partial\phi_i}{\partial x}+n_y\frac{1-\nu}{2}\frac{\partial\phi_i}{\partial y} & n_x\nu\frac{\partial\phi_i}{\partial y}+n_y\frac{1-\nu}{2}\frac{\partial\phi_i}{\partial x} \\ n_y\nu\frac{\partial\phi_i}{\partial x}+n_x\frac{1-\nu}{2}\frac{\partial\phi_i}{\partial y} & n_y\frac{\partial\phi_i}{\partial y}+n_x\frac{1-\nu}{2}\frac{\partial\phi_i}{\partial x} \end{array}\right]\left\{\begin{array}{c} u_i \\ v_i \end{array}\right\} = \left\{\begin{array}{c} \bar{t}_x \\ \bar{t}_y \end{array}\right\} \tag{7.69}$$

Compared to weak form based meshfree methods, the collocation methods are much more sensitive to the nodes included in the support domain. It appears that in general, increasing the number of nodes is a requirement to obtain good accuracy.

Figure 7.34 (right) shows the influence of the number of nodes in the support domain for the col-MLS on accuracy. As shown on figure 7.34 (left), it is necessary to use at least a cubic basis to obtain accurate results. Through our studies, it has been found that a cubic basis gives the best compromise for the Col-MLS between speed and accuracy. Augmenting the basis leads to the need to include more nodes in the support domain. On average the collocation method already

requires more nodes in the support domain to obtain accurate results than the weak formulation based methods. It has to be noted that when the basis is small or when insufficient nodes are within

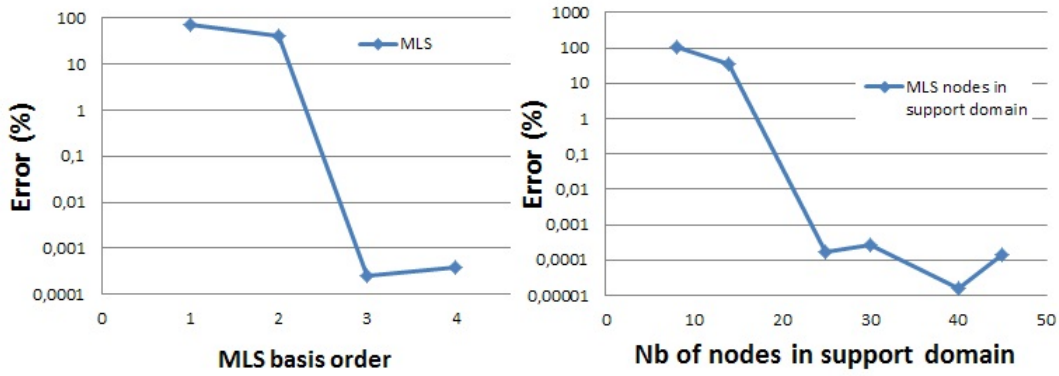


Figure 7.34: Influence on the approximation error of the number of nodes in the support domain (left) and the size of the basis(right)

the support domain, the moment matrix is ill conditioned with a condition number $\approx 8.3 \times 10^{25}$. This results in poor accuracy and a higher computational cost to build the shape functions. The increase in overall computational time is not significant when a proper domain size is set.
Surprising, col-MLS provides very good accuracy for small distribution of nodes between 12-400. This can be observed on figure 7.35 for regular set of nodes and on 7.36 for irregular nodal distribution The phenomenon of better accuracy for small distribution of nodes is much less pronounced for Col-RPIM. It can be observed however for the col-RPIM-EXP. However as the number of sampling point increases, the accuracy decreases. An increase of the support domain size does not impact this effect. It seems that the interpolant becomes less accurate as its support gets smaller with a denser set of data.

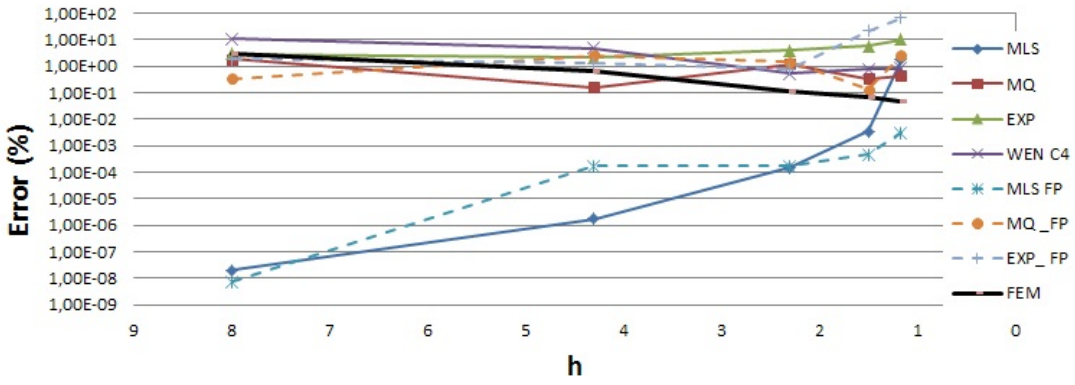


Figure 7.35: Comparison of the convergence for different collocation methods for regular node distribution

A more standard behaviour is observed for col-RPIM (see figure7.35 and 7.36. However the accuracy of the method is far inferior, which is the opposite of what we observed for Poisson/ heat conduction problems. This definitely does not help in choosing one of the method. For mix thermo-elastic problems, using the more efficient interpolant for each energy domain might produce a better overall result. For RPIM the increase in the number of nodes improves the accuracy as observed before. However too few or too many nodes in support domain degrades the accuracy of the solution.

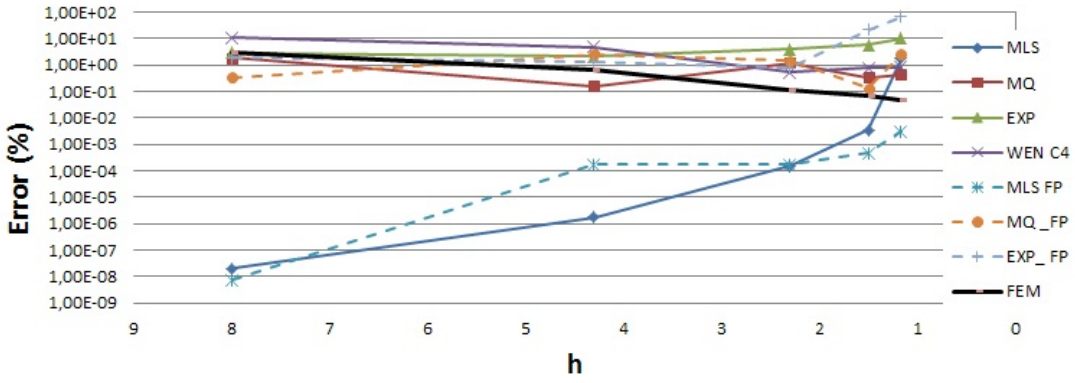


Figure 7.36: Comparison of the convergence for different collocation methods for irregular node distribution

Another feature is the capability for both method to work well for irregular nodal distribution as shown on figure7.36. This is especially the case for the col-RPIM where results are better for irregular nodal distribution than for regular ones where the accuracy tends to degrade when the density of nodes is too important. The collocation method seems to hilight the quality of the interpolant more than the weak formulation.
The choice of parameters is much more sensitive than for weak formulated methods. It has been found that the col-Wen-C4 works much better with polynomial enrichment, especially for irregular set of nodes. But no improvements for the MQ or EXP SFs were found through the experiment. As shown on figure 7.35 and 7.36, using FP does not provide better results for this problem, except for dense set of nodes.
Overall, better results are obtained for control support domain and when we use FP. The col-MLS surprisingly works very well. However col-RPIM works does not work so well and parameters need to be tuned differently than for previous problems. For instance, we found that $alpha_c \approx 6$ for MQ and $\alpha_c = 0.01$ for EXP seems to provide better results. Figure 7.37 shows a comparison of the computational time for the collocation methods with FEM using Quad bilinear element. Regular nodal distribution is used for the comparison.

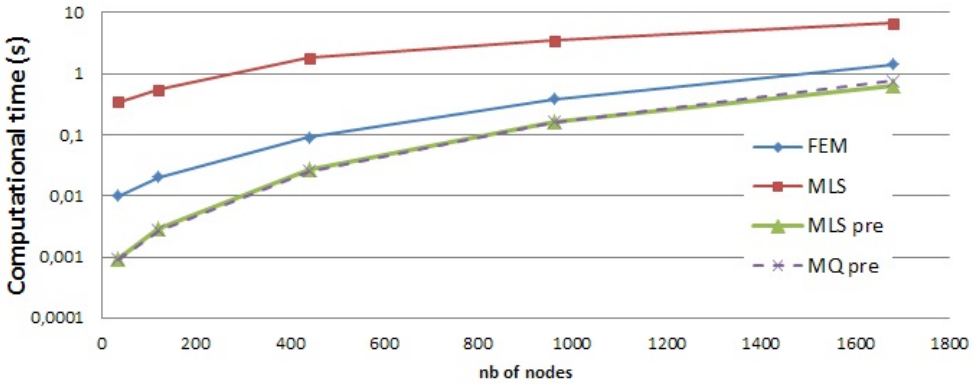


Figure 7.37: Comparison of the computational time for different collocation methods and FEM.

We can observe that FEM is faster than the collocation methods. The reason is that the number of nodes used to build the SF is much more important for collocation methods. However using precomputing at the nodes creation stages allows us to decrease the overall computational time

by a great magnitude. Because collocation based methods are truly meshless where nodes can be moved easily without the need to recompute all the Sfs. Hence using Precomputing of Sfs makes the collocation methods much faster than FEM as shown on figure 7.37 for MLS and MQ. As shown on figure7.35, the collocation method is also competitive in term of accuracy with FEM.

7.4.6 2D elasto-dynamic problem

Considering an isotropic materials, the strong form for dynamics analysis is given by equation 6.87 witch in 2D can be written

$$\begin{aligned}
&\frac{E}{1-\nu^2}\left(\frac{\partial^2 u}{\partial x^2}+\frac{1-\nu}{2}\frac{\partial^2 u}{\partial y^2}+\frac{1+\nu^2}{2}\frac{\partial^2 v}{\partial x \partial y}\right)+b_x-\rho\frac{\partial^2 u}{\partial t^2}-c\frac{\partial u}{\partial t}=0\\
&\frac{E}{1-\nu}\left(\frac{\partial^2 v}{\partial y^2}+\frac{1-\nu}{2}\frac{\partial^2 v}{\partial x^2}+\frac{1+\nu}{2}\frac{\partial^2 u}{\partial x \partial y}\right)+b_y-\rho\frac{\partial^2 v}{\partial t^2}-c\frac{\partial v}{\partial t}=0
\end{aligned} \tag{7.70}$$

Where ν, ρ, E and c are the Poisson's ratio, the mass density, the Young modulus and damping coefficient respectively.
Using collocation procedure and the meshfree approximation given by equation7.68 we can write the following discretised system of equations at each field node x_I, $I = 1...N$.

$$M_I\ddot{u}(t) + C_I\dot{u}(t) + K_I u(t) = F_I(t) \tag{7.71}$$

Where $u(t) = \{u_I v_I\}$ is the vector of nodal displacement. K_I is the same system matrix obtained in equation7.65 and M_I and C_I being respectively the nodal mass matrix and the nodal damping matrix written

$$M_I = -\rho\Phi_I; \quad C_I = -c\Phi_I \tag{7.72}$$

With Φ_I the vector of SFs evaluated at collocation node x_I. Applying this discretisation at all N nodes and stacking all this 2N nodal system of equation gives the final system equation of the problem

$$M\ddot{u}(t) + C\dot{u}(t) + Ku(t) = F(t) \tag{7.73}$$

The same forced vibration analysis problem found in section 6.6 is considered. Figure7.38 shows the results obtained for a pulse load by the Col-MLS using 121 irregular nodal distribution.

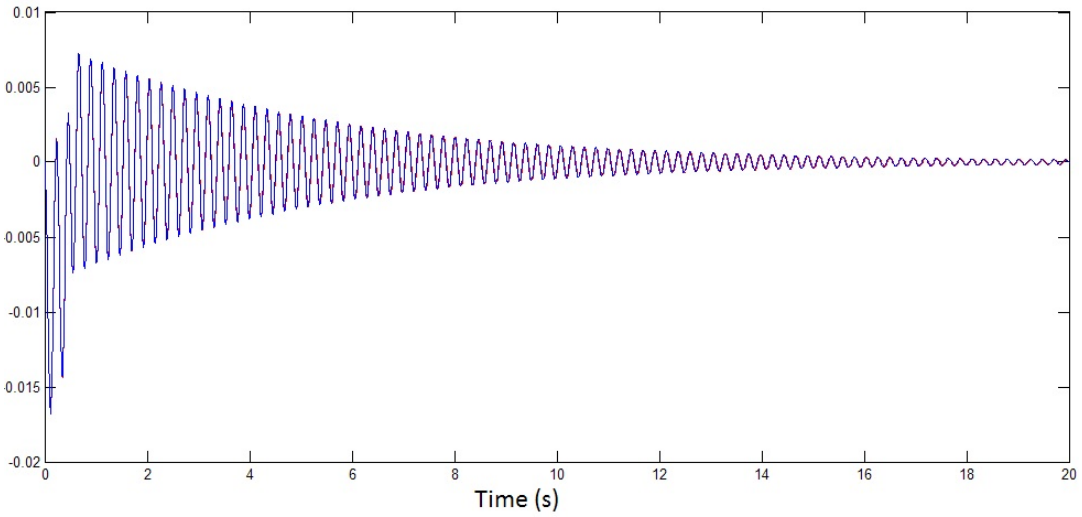


Figure 7.38: Solution of forced vibration problem using Col-MLS with

We obtained very accurate results using the MLS interpolant similar to the one from section 6.6

7.5 Conclusion

Compared to methods based on weak formulation the collocation method shows a number of advantages. Due to their strong-formulation, there is no need to perform numerical integration unlike for methods based on weak formulation. It results in an ease in the implementation of these methods. Collocation meshfree methods are computationally more efficient than their counterpart based on weak-form. Not requiring numerical integration and hence also the grid for the numerical quadrature provides these methods a true "meshfree" feature.

There are several issues encountered when using collocation methods. First the discretisation operates directly on the strong form of the equation problem. Unlike integration, which is a smoothing operator, differentiation is a roughening operator and usually magnifies errors in the approximation scheme. This is often the cause of instability in the solution. The other problem of collocation methods is the presence of DBCs. When the boundary conditions of the problem are all of Dirichlet type, the collocation method produces accurate and stable results both for regular and irregular sets of nodes. However, presence of the Neumann or robin boundary conditions can deteriorate the quality of the solution in a significant manner causing it to become unstable. When DBCs are discretised using direct collocation method to obtain the separate set of equations, the discretised system of equation can behave like an ill-posed problem where a small input error can become dramatically amplified.

The issue lies in the presence of the derivative of the field function in the boundary condition (DBC). How well the DBC is imposed greatly determines highly the accuracy of the method.

Several techniques [162] can be used to partially alleviate the problem. Among them, the fictitious point method provides a cheap solution in terms of computational charge but it does not always work well. Methods based on adding derivatives at the DB nodes in the basis of the shape function are another solution. One main drawback lies in the use of list and search algorithm among this dynamic list of nodes. For instance in HCM, one has to look for the presence of boundary nodes within the nodes list that corresponds to one support domain in order to enrich the basis with the derivative of the field at these nodes. This class of methods also complicates the building of SFs

resulting in an increase of the overall computational cost.
Lastly mixing the strong form with local weak form to use the advantage of the natural way of implementing the Neumann conditions in the local weak form is probably the most efficient [166]. However it brings back the problem inherent to local weak form which are the cost of integration and local support domain management.
Methods to stabilise collocation methods like adding higher order differential terms in the strong form equations have also been developed [209, 210]. However the error cannot always be controlled and a suitable technique might work for one problem but not for another of the same type.
Overall, compared to weak formulation based methods, the collocation ones are more unstable. We found them less accurate. RPIM works well for heat conduction problem whereas MLS better for the elastic application. The issue is that the strong form requires a higher order derivative of the shape function. This inevitably results in some added errors as the accuracy of the interpolation drop with the increase of the derivatives order. On the other hand, weak formulation tends to smoothen the error and require Sfs derivative of one less order than the PDE one. The conclusion is that collocation is less robust and unstable. We also found that collocation methods require much attention regarding the choice of parameters. Once again the Weak formulation is more forgiving on this aspect. The size of support domain is α_s is one example. In EFG, the generic range can be defined for the whole range of applications and rarely degrades the quality of the results in a dramatic manners. This is not the case for collocation methods and the difference between un-optimised and optimised support domain size is important. For some problems like the non-linear one, it is one of the primary criteria.
Nevertheless, strong-form based MF methods have a huge advantage, which is the absence of integration. In most cases despite the instability, when the parameters are set right, the methods work very well with irregular nodal distribution. They are faster than both global and local weak formulation based methods and even the FEM. The most computationally demanding part remains in the MF SFs calculation. In this aspect, they are more costly than for weak formulation based methods because they require their derivative up to the order of the PDE and usually more nodes in the support domain to obtain quasi similar accuracy.
Although benchmarks for complicated shapes were not covered in this thesis, applying the methods for simple benchmarks highlighted the capability of the methods to handle irregular nodes well with no difficulty in implementation. Hence, due to the absence of the need for an integration support structure and that there is no requirement on nodal location, the adaptivity to complex shape can be deduced. However an implementation for complex MEMS multi-structure with several material layers would be necessary to validate collocation methods as a potential solution.

Chapter 8 Conclusion and future work

8.1 Conclusion

In this thesis, we have presented the concept of using haptic technology for thermal management in the design of MEMS. The first part is devoted to the study of the specifications and the implementation of a demonstrator for the haptic thermal management of MEMS design tool. A solution in the form of haptic texturing for thermal feedback through a haptic channel provided encouraging results. The study of CD constraints and the requirement for an accurate physical model has pushed us to select a mesh based model such as FEM and FDM to render simulated temperature field and induced body deformations. The implementation of a demonstrator has highlighted the limitations in the use of regular mesh model such as FDM or FEM for the simulation of the physics of the model for real time haptic interaction.

To circumvent these limitations, the second part of the thesis was devoted to exploring a new methodology to model MEMS physics component in the VE. Meshfree methods were targeted as a potential solution. A careful review of mesh free methods and benchmark tests were conducted to determine the potential of these methods for incorporation in haptic, and to solve the mesh limitation for the update rate. Other potential benefits such as a better flexibility to conform to complex domains for haptic deformations and shape modifications were targeted. Table 8.1 page 203 gives a summary of the main features of meshfree methods compared to FEM.

From our studies, meshfree methods provides good results in term of accuracy compared to the standard FEM. Weak formulation based methods are more accurate but are computationally more costly, whereas collocation methods are faster but unstable. Overall, except for certain applications such as moving boundaries or topology modifications, these methods do not provide benefits over the standard FEM. In our case it could benefit for the thermal exploration with cutting process and isothermal drawing and rendering. Based on the features of the different methods, we propose a compromised solution based on a multi-layer thread which should be implemented using the collocation method for the real-time thermal modelling. A loosely coupled techniques with a spring model and mix collocation-local weak form for the elastic deformation induced by the thermal stress, could speed up the computation and ensure accuracy.

Through the work carried out in this thesis, the use of meshfree methods as a solution for the physical model of objects in VE does not appear justified anymore. One of the main reasons is that meshes provide too much of a structure for other parts of the system such as graphical rendering and collision detection. Meshfree methods simply do not provide enough benefits to move away from the use of the mesh for the virtual model analysis. Another issue with meshfree techniques is the difficulty in treating multi-layer materials and structures, which forms the basis for MEMS design. The thermal behaviour of laminated materials is of prime importance in design as well as for mitigating failure such as delamination in backing process. Contrary to FEM where the mesh provides a structure to support material properties through the geometry, in the case of meshfree

methods, the purely nodal approximation based on area of influence brings more difficulty and requires special treatment for the implementation of multi-layer materials.
FEM is now a mature technology considered by the engineering community and the industry as the standard simulation and design tool. On the other hand, meshfree methods are still considered to be in an infancy stage, and in the past 20 years, much effort has been dedicated to making this technique reliable for engineering commercial package applications. With the appearance of enrichment methods such as PUFEM, XFEM and GFEM [14,183], the need for meshfree methods has been questioned in the research community and the effort invested to make them robust has significantly dwindled. The partition of unity allows the enrichment with special functions as it can be done in the basis with EFG, releasing the dependency on the mesh imposed by regular FEM for cracks or singularities problems. However they are constructed over regular FEM mesh.

8.2 Future work

The rapid advance in parallel computing and general-purpose processing on graphics processing units (GPGPU) computing solution has allowed to make great breakthroughs in real time simulation and cutting interaction with deformable soft tissue for virtual surgery tools [42, 67, 68]. The model can rely on either a mesh based or a meshless method. Implementation of meshfree models using efficient GPGPU acceleration solution such as CUDA C++ combined with parallel architecture should be able to help with satisfying the real time update criteria in our haptic application.
Another direction is to use iso-geometric methods(IGM) [126] which rely on the same information the CAD system uses to create geometry for analysing fields such as the temperature and displacement. IGMs use NURBS [258], T-splines [22] or hierarchical THB-splines [137] which are the cornerstone for the mathematical representation of shape in modern CAD system. A good overview of iso-geometric analysis methods(IGAMs), benefits and the issues with implementation can be found in [202]. Iso-geometric BEM [251,262] requires only a boundary discretisation which greatly reduces computational cost and can circumvent mesh generation by using CAD discretisation for analysis. From an industrial standpoint the coupling of the CAD model and the physics model is a great feature for virtual prototyping. This allows for significant time reduction in the design process cycle and optimization problems [180,198,290].
In our case, building approximations using IGAM with surface discretisation would allow a unification of the physic model, the CD and the force rendering, hence merging both CAD and the simulation process in a haptic environment. For these reasons, IGAM combined with GPU computing seems to be a very appealing solution to model the virtual prototype in our haptic tool for MEMS design and simulation.

Table 8.1: Comparison of FEM, EFG and MLPG and collocation's features

	FEM	EFG	MLPG	Collocation methods
Formulation	Global weak form	Global weak form	Local weak form	Strong form
Nodes data structures	Nodes coordinates, mesh connectivity	Nodes coordinates and connectivity structure	Nodes coordinates and element's connectivity	Nodes coordinates
subdomain	Element based, polygonal types	Spherical or quadrilateral. Overlapping. Polygonal type/ Voronoi cell.	Any type.	any type
Shape functions	Direct form. polynomial	No closed form, complex. Non polynomial and non rational function.		
Integration	Low order	Moderate-high order, Costly Moderate for nodal integration	High order very .Costly	No need
Essential bcs	Direct , straightforward.	MLS,PUM,Hp: special treatment. Costly specially for 3D RPIM,PIM: direct, straightforward	Direct, easy	Direct straightforward
System matrix	Symmetric, banded, positive definite.	Symmetric, banded. Possibly badly conditioned	Asymmetric, banded Possibly badly conditioned	Asymmetric, banded instability
Assembly procedure	Element based. Straightforward	Node based.	FDM stacking type. Straightforward	
Addition of nodes	Complex. Re-meshing	easy with background grid, complex for cell integration	very easy	Very easy
Treatment of material discontinuity	Easy nodes based.	Complex. Require the use of special basis functions, . Lagrange multipliers or jump function		
Data pre-process	Mesh generation.	Nodes management, neighbours search. through bucket or tree. Mildly complex		
Boundary compliant	Easy to extremely complex Yes. Difficult for complex bds.	Yes	Yes, costly. Need to determine subdomain intersection with bds	Yes
Singularities, cracks, Discontinuities	Very complex.	reasonably complex. More efficient with enriched basis and special basis functions		
Stress, strain , energy.	Heavy load on meshing. Discontinuous at element intersection. Usually linear within element.	Smooth throughout the domain. More accurate than FEM		less accurate than weak formulation
Multi-physics, multi scale	Very complex. Computationally demanding.	Promising	promising/ integration through collocation required to control computational cost.	?
Adaptivity	Complex. Estimate available. Costly re-meshing procedure	Few energy estimate available. Cell based. Open problem	None.	?
Stability	stable	stable	depend on integration and subdomain	rather unstable
Commercial	Yes. Mature technology.	Few	Seldom	?

Appendix A Weight residual method

The weighted residuals method (WRM) involves the approximation of the functional behavior of the dependent variable in the governing differential equation [83, 122, 159, 307]. When substituted into the governing differential equation, the approximate form of the dependent variable leads to an error called the "Residual". This residual error is required to vanish in a weighted average sense over the domain. If the weighting functions are chosen to be the same as the element shape (interpolation) functions used in the element approximation functions, the method of weighted residuals is referred as the Galerkin's method.
Let us consider a general problem over a domain Ω and its boundary $\partial\Omega = \Gamma$ defined by the following partial differential equation

$$\begin{aligned} &L(u(x)) - f(x) = 0, \quad \forall x \in \Omega \\ &G(u(x)) = g(x), \quad \forall x \in \Gamma \end{aligned} \tag{A.1}$$

Where L and G are differential (We can restrain to partial differential operators for our applications) operators. F and g being given functions. We suppose that the approximate solution is of the finite series form

$$\hat{u} = \sum_{i=1}^{N} \Phi_i a_i \tag{A.2}$$

Where Φ_i are the basis functions also called trial function, a_i the unknown coefficients and N the number of basis functions used. The basis functions should satisfy conditions such as admissibility , which refers to their capability to satisfy the essential boundary conditions and other continuity requirement. Usually, $\hat{u}$ does not satisfy the differential equation together with the boundary conditions A.1

$$L(\hat{u}(x)) - f(x) \neq 0, \quad \forall x \in \Omega G(\hat{u}(x)) - g(x) \neq 0, \quad \forall x \in \Gamma \tag{A.3}$$

We can define the residuals

$$R_s(\hat{u}) = L(\hat{u}(x)) - f(x) \neq 0, \quad \forall x \in \Omega R_b(\hat{u}) = G(\hat{u}(x)) - g(x) \neq 0, \quad \forall x \in \Gamma \tag{A.4}$$

Then the residual is forced to be zero in an average sense by setting the weighted integrals of residuals to zero over the domain and boundaries domain.

$$\int_{\Omega} W_i(x) R_s(\hat{u}) d\Omega + \int_{\Gamma} V_i(x) R_b(\hat{u}) d\Gamma = 0 \tag{A.5}$$

W_i and V_i are called weight functions for the corresponding residual. In the case where the approximate solution is chosen to automatically satisfy the boundaries conditions the system A.5

simplifies into

$$\int_{\Omega} W_i(x) R_s(\hat{u}) = 0 \tag{A.6}$$

Using the equation A.2, we can expand the system of equations A.6, which gives a system of n equations of n unknowns a_i

$$\left.\begin{array}{c} \int_{\Omega} W_1(x) \left[L(\hat{u}(x)) - f(x)\right] d\Omega + \int_{\Gamma} V_1(x) \left[G(\hat{u}(x)) - g(x)\right] d\Gamma = 0 \\ \int_{\Omega} W_2(x) \left[L(\hat{u}(x)) - f(x)\right] d\Omega + \int_{\Gamma} V_2(x) \left[G(\hat{u}(x)) - g(x)\right] d\Gamma = 0 \\ \vdots \\ \int_{\Omega} W_N(x) \left[L(\hat{u}(x)) - f(x)\right] d\Omega + \int_{\Gamma} V_N(x) \left[G(\hat{u}(x)) - g(x)\right] d\Gamma = 0 \end{array}\right\} \tag{A.7}$$

The approximate solution makes the residuals vanish. In the case where the residual would be null, the approximated solution would be the exact solution of the problem. The weight residual method transforms PDE into a set of integral form equations. This form is equivalent in a certain way to the functional obtain through variational formulation. Equation A.5 is called the discrete weak formulation which can be seen as the discrete part of the weak formulation of the continuous problem. The solution corresponding to this weak formulation is called the weak solution of the problem A.1. This is because the formulation weakens the continuity order requirement on the approximate function by moving part of it on the weight function using integration by parts. The approximate solution will converge to the exact solution (unique and continuous) of the problem when $n \to \infty$ if W_i and the basis functions Φ_i are linearly independent and if the basis function Φ_i is smooth enough. The integral formulation is interesting in the way that it smoothes the error resulting from approximation of finite series formulation and also helps in terms of stability and accuracy [83, 162]. The choice of the weight function affects greatly the performance of the method.

Appendix B Weak formulations

B.1 Weak formulation for 2D heat conduction

We consider the 2 dimensional Poisson equation over the domain Ω with its boundary $\partial\Omega$.

$$k\nabla^2 u - f(x) = 0; \tag{B.1}$$

Together with the boundary conditions

$$u(x,y) = \bar{u}(x,y) \quad \forall x \in \Gamma_u \tag{B.2}$$

$$\frac{\partial u(x,y)}{\partial n} = u_{,i} n_i = \bar{q} \cdot n \quad \forall x \in \Gamma_q \tag{B.3}$$

In the case of heat conduction, equation B.2 corresponds to the Dirichlet boundary condition type where $\bar{u}$ is the temperature field function imposed directly at the boundary. Equation B.3 is the Neumann or natural type of boundary condition with $\bar{q}$ being the heat flux function. $f(x)$ is the internal heat generation source. Γ_u and Γ_q are segments of the boundary $\partial\Omega$ such that $\Gamma = \Gamma_q \cap \Gamma_u$. n_i represents the components in the Cartesian coordinates(x,y) of the unit normal $\vec{n}$ to Γ_2.
The weak formulation can be derived using WRM or the energy principle. For simplicity we assume that the medium is isotropic (k is a scalar). Using the WRM on the Poisson equation leads to

$$\int_\Omega \left[k\nabla^2 u - f(x)\right] v d\Omega = 0 \tag{B.4}$$

With v weight function. Integrating by part gives

$$\int_\Omega \nabla(vk\nabla u) d\Omega + k\int_\Omega \nabla v \nabla u = \int_\Omega v f(x) d\Omega \tag{B.5}$$

Using the divergence theorem gives rise to the symmetric weak formulation of the 2D Poisson equation

$$\int_{\partial\Omega} vk\nabla u \cdot n d\Omega + k\int_\Omega \nabla v \nabla u = \int_\Omega v f(x) d\Omega \tag{B.6}$$

The weak formulation makes the treatment of Neumann type boundary condition B.3 straightforward. Substituting this into the first term of equation B.6 leads to

$$\int_{\partial\Omega} v\bar{q} \cdot n d\Omega + k\int_\Omega \nabla v \nabla u = \int_\Omega v f(x) d\Omega \tag{B.7}$$

It is termed weak form as it smoothens the requirement on the trial function and moves it upon the

test function. The constrained Galerkin form using penalty method is given as follows

$$\int_{\Omega} \delta(\nabla u)^T \nabla u d\Omega - \int_{\Omega} (\delta u)^T f d\Omega - \int_{\Gamma_q} \bar{q} \delta u d\Gamma + \frac{\alpha}{2} \delta \int_{\Gamma_u} (u - \bar{u})^T (u - \bar{u}) d\Gamma = 0 \tag{B.8}$$

$\alpha = (\alpha_1, \alpha_1, \cdots, \alpha_1)$ is the diagonal matrix containing the penalty factor.

B.2 Weak formulation for the elasto-static problem

The strain energy is expressed as

$$U = \frac{1}{2} \int_{\Omega} \sigma_{ij} \epsilon_{ij} d\Omega \tag{B.9}$$

A more general form exist in the case of nonlinear behavior where the strain tensor field has to be expressed in term of increment depending on the type of non-linear behavior of the material [112]. The integration of the plastic increment has to be included in the formulation

$$U = \int_{\Omega} \int_{\epsilon} \sigma d\epsilon d\Omega \tag{B.10}$$

We express the work done by external force by

$$W = \sum_i F_i u_i = \int_{\Omega} f^b u^T d\Omega + \int_{\Gamma} f^t u^T d\Gamma \tag{B.11}$$

With f^b being the internal body force and f^t being the surface traction force. If the applied force in B.11 is conservative, then equation defines the negative value of potential energy generated by the applied loads. The applied load is considered conservative if it is independent of deformation, so that the work done by a system of applied forces in traversing any closed path in displacement space has to be zero. The total potential energy Π of the structure is the difference between the strain energy and the work done by the applied loads

$$\Pi = U - W \tag{B.12}$$

We consider an arbitrary virtual change in displacement δu, using the concept of variation of functional (stationary point) to the equilibrium equation to reformulate the static equilibrium equation B.2

$$\int_{\Omega} \left[\sigma_{ij,j} + b_i\right] \delta u d\Omega = 0 \tag{B.13}$$

Using integration by part gives

$$-\int_{\Omega} \sigma_{ij} \delta u_{i,j} d\Omega + \int_{\Omega} b_i \delta u_i d\Omega + \int_{\Gamma} \sigma_{ij} n_j \delta u_i d\Gamma \tag{B.14}$$

Where n_j is the normal outward to the boundary surface Γ. Using the traction boundary condition, we obtain

$$-\int_{\Omega} \sigma_{ij} \epsilon_{ij} d\Omega + \int_{\Omega} b_i \delta u_i d\Omega + \int_{\Gamma_1} \bar{t}_i \delta u_i d\Gamma = 0 \tag{B.15}$$

Now if we consider the virtual work based on equation B.11 using the concept of variation of

functional due to the virtual change in displacement δu

$$\delta W = \int_{\Omega} f_i^b \delta u_i d\Omega + \int_{\Gamma} f_i^t \delta u_i d\Gamma \tag{B.16}$$

Using B.15, choosing the virtual displacement kinematically admissible $\delta u = 0, \quad \forall x \in \Gamma_2$ we can write using the Gauss theorem,

$$\delta W = \int_{\Omega} b_i \delta u_i d\Omega + \int_{\Gamma_1} t_i \delta u_i d\Gamma = \int_{\Omega} \sigma_{ij,j} + b_i \delta u d\Omega + \int_{\Omega} \sigma_{ij} \epsilon_{ij} d\Omega \tag{B.17}$$

From equation B.13, the first term on the right hand side is vanishing and leads to the following form for the virtual work

$$-\int_{\Omega} \sigma_{ij} \epsilon_{ij} d\Omega + \int_{\Omega} b_i \delta u_i d\Omega + \int_{\Gamma_1} t_i \delta u_i d\Gamma = 0 \tag{B.18}$$

The first term on the left hand side of this equation can be interpreted as the virtual work done by the internal stresses δW_I. We can group the other terms under the work done by the external applied force and surface traction δW_E. Then, we simply obtain the following form for the virtual work equation

$$\delta W = \delta W_I + \delta W_E = 0 \tag{B.19}$$

The virtual work principle, by reversing the integration process implies equation B.15 and B.13 which must hold for kinematically admissible virtual displacements. This equation and the principle of virtual work hold only if the stress field is statically admissible.

B.3 Weak formulation of elasto-dynamic problem

For a system of elastic solids and structures the Lagrangian functional can be defined as the sum of the kinetic energy T, the strain energy Π and the work done by external forces W_f over the whole domain Ω.

$$L = T - \Pi_s + W_f \tag{B.20}$$

The kinetic energy T is expressed as

$$T = \frac{1}{2} \int_{\Omega} \rho \dot{u}^T \dot{u} d\Omega \tag{B.21}$$

For solids and structures of elastic materials, the strain energy of the system can be expressed as

$$\Pi_s = \frac{1}{2} \int_{\Omega} \epsilon^T \sigma d\Omega \tag{B.22}$$

And the work done by external forces can be obtained by

$$W_f = \int_{\Omega} u^T b d\Omega + \int_{\Gamma_t} u^T t d\Gamma \tag{B.23}$$

Γ_t is the boundary of the solid on which traction forces t are operating.
Consider the structural elastic problem. For a time dependent load l the response of the system will be time dependent. With the particle moving, velocity $u_{,t}$ created results in the generation of

kinetic energy T. Taking the first variation of it gives

$$\delta T = \frac{1}{2}\int_{\Omega} \rho \delta u_{,t}^{T} u_{,t} d\Omega \tag{B.24}$$

Now to reformulate this equation in terms of virtual displacement, we consider Kinematically admissible displacement such that $\delta u(x,0) = \delta u(x,t_f) = 0$. Now integrating equation B.24 over the time interval gives

$$\int_{t_i=0}^{t_f} \delta T dt = \int_{t_i=0}^{t_f} \frac{1}{2}\int_{\Omega} \rho \delta u_{,t}^{T} u_{,t} d\Omega dt \tag{B.25}$$

Integrating by part and using the condition on displacement gives

$$\int_{t_i=0}^{t_f} \delta E_k dt = -\int_{t_i=0}^{t_f} \frac{1}{2}\int_{\Omega} \rho \delta u^{T} u_{,tt} d\Omega dt = -\int_{t_i=0}^{t_f} d(u_{,tt}, \delta u) dt \tag{B.26}$$

With $d(u_{,tt}, \delta u)$ being the kinetic energy bilinear form. Now considering a system under conservative load and using the Lagrangian principle of least action

$$\delta \int_{0}^{t_f} \left[\Pi(z) - T(z_{,t})\right] dt = 0 \tag{B.27}$$

Rewriting this equation using the bilinear form or variational form for each quantity leads to

$$\delta \int_{0}^{t_f} \left[a(u,\delta u) - l(\delta u) - d(u_{,tt}, \delta u))\right] dt = 0, \;\; \forall u \in E. \tag{B.28}$$

Lagrangian principle of least action provides the variational equations of structural dynamics.

References

[1] T. Akin. CMOS-based thermal sensors. *Advanced Micro and Nanosystems CMOS-MEMS*, Vol 2, 2005.

[2] S.N. Alturi. *Methods of computer modeling in engineering and the sciences Vol1*. CREST. Tech Science Press, USA, 2005.

[3] S.N. Alturi, Z.D. Han, and A.M. Rajendran. A new implementation of the meshless finite volume method, through the MLPG "mixed" approach. *CMES*, 6(6):491–513, March 2004.

[4] S.N. Alturi and S. Shen. The meshless local Petrov-Galerkin (MLPG) method: A simple and less-costly alternative to the finite element and boundary element methods. *CMES*, 3(1):11–51, 2002.

[5] S.N. Alturi and S. Shen. The basis of meshless domain discretization: the meshless local Petrov-Galerkin (MLPG). *Advances in Computational Mathematics*, 23:73–93, March 2005.

[6] S.N. Alturi and T. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational mechanics*, Vol 22:117–127, 1998.

[7] S.N. Alturi and T. Zhu. A critical assessment of the truly meshless local Petrov-Galerkin method (MLPG), with generalized moving least squares interpolation. *Computational mechanics*, 24:334–347, 1999.

[8] N.R Aluru. A point collocation method based on reproducing kernel approximations. *International journal for numerical methods in engineering*, 47:1083–1121, 2000.

[9] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *In Eurographics 87*, pages 3–10, 1987.

[10] W.F. Ames. *Numerical Methods for Partial Differential Equations*. Acadamic Press, New York, 1992.

[11] M. A.Puso, J. S. Chen, E. Zywicz, and W. Elmer. Meshfree and finite element nodal integration methods. *International Journal for Numerical Methods in Engineering*, Vol 74(3):1097–0207, 2008.

[12] G. Awcock. Principles of image data compression eo307 digital signal processing modeule. Technical report, Univerity of Brighton, 2000.

[13] I. Babuska. The finite element method with Lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.

[14] I. Babuska, U. Banerjee, and J.E. Osborn. Meshless and generalized finite element methods: A survey of some major results. *Acta Numerica*, 12:1–125, February 2003.

[15] I. Babuska and J.M. Melenk. The partition of unity method. *International Journal of Numerical Methods in Engineering*, 40:727–758, 1996.

[16] D.S. Balsara. von Neumann stability analysis of smoothed particle hydrodynamics–suggestions for optimal algorithms. *Journal of Computational Physics*, 121(2):357 – 372, 1995.

[17] P.K. Banerjee. *The boundary element methods in engineering*. McGraw-Hill, new-York, 1994.

[18] C. Basdogan. Force-reflecting deformable objects for virtual environments. Technical report, 1999.

[19] C. Basdogan, C.H. Ho, and M.A. Srinivasan. A ray-based haptic rendering technique for displaying shape and texture of 3d objects in virtual environments. In *Proc. ASME Dynamic Systems and Control Division*, pages 77 – 84, 1997.

[20] C. Basdogan, S. D. Laycock, A. M. Day, V. Patoglu, and R. B. Gillespie. Haptic rendering. In *In Virtual Environments, Handbook of Virtual Environments*, pages 117–134. Lawrence Earlbaum, Inc, 2002.

[21] Bassens. http://thalamus.wustl.edu/course/bassens.html.

[22] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(58):229 – 263, 2010. Computational Geometry and Analysis.

[23] T. Bechtold, E.B. Rudnyi, and J.G. Korvink. Dynamic electro-thermal simulation of microsystems,a review. *Journal of Micromechanics and Microengineering*, 15(11):1–7, 2005.

[24] A.A. Becker. *The boundary element methods in engineering*. McGraw-Hill, London, 1994.

[25] E.B. Becker, G.F. Carey, and J.T. Oden. *Finite Element methods: An introduction Volume 1*. Prentice-Hall, Englewood Cliffs, NJ, 1981.

[26] S. Beissel and T. Belytschko. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):49 – 74, 1996.

[27] T. Belytschko, Y. Guo, W.K Liu, and X.S Ping. A unified stability analysis of meshless particle methods. *International Journal for Numerical Methods in Engineering*, 48(9):1097–0207, 1999.

[28] T. Belytschko, Y. Krongauz, J. Dolbow, and C. Gerlach. On the completeness of meshfree particle methods. *Int. J. Numer. Mech. Engng*, 43:785–819, 1998.

[29] T. Belytschko, Y. Krongauz, M. Fleming, D. Organ, and W.K.S. Liu. Smoothing and accelerated computations in the element free Galerkin method. *Journal of Computational and Applied Mathematics*, 74(1-2):111 – 126, 1996.

[30] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless methods: “ an overview and recent developments”. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1996.

[31] T. Belytschko, Y.Y. Lu, and L. Gu. Crack propagation by element-free Galerkin methods. *Engineering Fracture Mechanics*, 51(2):295 – 315, 1995.

[32] T. Belytschko, Y.Y. Lu, L. Gu, and M. Tabbara. Element-free Galerkin methods for static and dynamic fracture. *International Journal of Solids and Structures*, 32(17−18):2547 – 2570, 1995. A Symposium on the Dynamic Failure Mechanics of Modern Materials In Memory of Professor J. Duffy.

[33] W. Benz. Applications of smooth particle hydrodynamics (sph) to astrophysical problems. *Computer Physics Communications*, 48(1):97 – 105, 1988.

[34] W. Benz. Smooth particle hydrodynamics - a review. In J.R. Buchler Editor, editor, *Proceedings of the NATO Advanced Research Workshop on The Numerical Modelling of Nonlinear Stellar Pulsations Problems and Prospects*, page 269, held in Les Arcs, France, March 20-24 1990. Publisher, Kluwer Academic Publishers, Dordrecht, The Netherlands.

[35] M. De Berg. *Computational geometry: algorithms and applications : with 370 illustrations*. Springer Verlag, 2000.

[36] P. Bertczick, G. Hensler, and C. Theis. Chemodynamical modelling of galaxy formation and evolution. *Journal of Astrophysics and Space Science*, 281:297–300, 2002.

[37] Daniel Bielser and Markus H. Gross. Interactive simulation of surgical cuts. In *Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, PG 00, page 116, Washington, DC, USA, 2000. IEEE Computer Society.

[38] S.J. Biggs and M.A. Srinivasan. Haptic interfaces. Technical report, Laboratory for human and Machine Haptics(the touch lab). MIT.

[39] J. Blauert, H. Lehnert, J. Sahrhage, and H. Strauss. An interactive virtual-environment generator for psychoacoustic research. i: Architecture and implementation. *Acta Acustica united with Acustica*, 86(1):94–102, 2000.

[40] J. Bonet and S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *Comput.mehtods Appl. Mech. Engrg*, 47(6):1189–1214, 2000.

[41] J. Bonet and S. Kulasegaram. Finite increment gradient stabilization of point integrated meshless methods for elliptic equations. *Communications in Numerical Methods in Engineering*, 16(7):475–483, 2000.

[42] J. Bosman, C. Duriez, and S. Cotin. Connective Tissues Simulation on GPU. In Jan Bender, Jeremie Dequidt, Christian Duriez, and Gabriel Zachmann, editors, *Workshop on Virtual Reality Interaction and Physical Simulation*. The Eurographics Association, 2013.

[43] C.A. Brebbia, J.C.F. Telles, and L.C. Wrobel. *The boundary element techniques*. Springer, Berlin, 1983.

[44] S.C Brenner and L.R Scott. *The mathematical theory of finite element methods*. Springer, 2002.

[45] S.C. Brenner and L.R. Scott. The construction of a finite element space. In J. E. Marsden, L. Sirovich, and S. S. Antman, editors, *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*, pages 69–92. Springer New York, 2008.

[46] Morten Bro-Nielsen and Ste S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum*, 15(3):57–66, 1996.

[47] C. Bruyns, S. Senger, S. Wildermuth, K. Montgomery, and R. Boyle. Real-time interactions using virtual tools. In *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, MICCAI '01, pages 1349–1351, London, UK, UK, 2001. Springer-Verlag.

[48] M. Buhmann and N. Dyn. Spectral convergence of multiquadric interpolation. *Proceedings of the Edinburgh Mathematical Society (Series 2)*, 36:319–333, 6 1993.

[49] M.D. Buhmann. Radial functions on compact support. *Proceedings of the Edinburgh Mathematical Society (Series 2)*, 41:33–46, 2 1998.

[50] M.D. Buhmann. A new class of radial basis functions with compact support. *Mathematics of Computation*, 70:307–318, january 2001.

[51] M.D. Buhmann. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA, 2003.

[52] H. K. Cakmak and U. Kuhnapfel. *Animation and Simulation Techniques for VR-Training Systems in Endoscopic Surgery*, pages 173–185. Springer Vienna, Vienna, 2000.

[53] M. Calis. Haptic sensing technologies for mems design and manufacture. Technical report, Heriot-Watt University,MicroSystems Engineering Centre (MISEC), 2006.

[54] P. Campbell. Some new algorithms for boundary value problems in smooth particle hydrodynamics. Technical report, Mission research Corporation, 1989.

[55] J. Chaskalovic. Variational formulations. In *Finite Element Methods for Engineering Sciences*, pages 63–111. Springer Berlin Heidelberg, 2008.

[56] J-S. Chen, P. Chunhui, C-T. Wu, and W.K. Liu. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):195 – 227, 1996.

[57] J-S. Chen, C-T. Wu, S. Yoon, and You. Yang. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):1097–0207, 2001.

[58] J-S. Chen, S. Yoon, and C-T. Wu. Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 53(12):1097–0207, 2002.

[59] K.K. Choi and N.H. Kim. Variational methods of structural systems. In *Structural Sensitivity Analysis and Optimization 1*, Mechanical Engineering Series, pages 37–62. Springer New York, 2005.

[60] A.J. Chorin. Numerical study of sligthly viscous flow. *Journal fo fluid mechanics*, 57(4):785–796, 1973.

[61] A.J. Chorin. Vortex models and boundary layer instability. *SIAM J.sci and Stat comput*, 1(1):1–21, March 1980.

[62] P.G. Ciralet and J.L. Lions. *Handbook of numerical analysis: Finite difference methods:*, volume Vol 1. North-Holland, 1991.

[63] P.G. Ciralet and J.L. Lions. *Handbook of numerical analysis: Finite element methods:part 1*, volume Vol 2. North-Holland, 1991.

[64] P.W. Cleary and J.J. Monaghan. Conduction modelling using smoothed particle hydrodynamics. *Journal of Computational Physics*, 148(1):227 – 264, 1999.

[65] J.D. Cohen, M.C. Lin, D. Manocha, and M.K. Ponamgi. I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments. In *Symposium on Interactive 3D Graphics*, pages 189–196, 218, 1995.

[66] S. Cotin, H. Delingette, and N. Ayache. *Real time volumetric deformable models for surgery simulation*, pages 535–540. Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.

[67] S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):62–73, January 1999.

[68] Hadrien Courtecuisse, Jérémie Allard, Pierre Kerfriden, Stéphane PA Bordas, Stéphane Cotin, and Christian Duriez. Real-time simulation of contact and cutting of heterogeneous soft-tissues. *Medical image analysis*, 18(2):394–410, 2014.

[69] Coventor. http://www.coventor.com/mems-solutions/products/coventorware/.

[70] W. Dai and T. Niu. A finite difference scheme for solving a nonlinear hyperbolic two-step model in a double-layered thin film exposed to ultrashort-pulsed lasers with nonlinear interfacial conditions. *Nonlinear Analysis: Hybrid Systems*, 2(1):121 – 143, 2008.

[71] Dassault7x. http://www.dassaultfalcon.com/en/Technology/Innovation/Pages/PLM-and-PLM-system.aspx.

[72] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. Dynamic Real-Time Deformations using Space and Time Adaptive Sampling. In *Computer Graphics Proceedings*, Annual Conference Series, Los Angeles, California, United States, 2001. ACM Press / ACM SIGGRAPH.

[73] H. Delingette, G. Subsol, S. Cotin, and J. Pignon. Craniofacial surgery simulation testbed. In R. A. Robb, editor, *Visualization in Biomedical Computing 1994*, volume 2359, pages 607–618, September 1994.

[74] J. Dolbow and T. Belytschko. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics*, 23:219–230, 1999.

[75] C.A. Duarte. A review of some meshless methods to solve partial differential equations,tech.report95-06. Technical report, TICAM,University of Texas, East Lansing, Michigan, 1996.

[76] C.A. Duarte and J.T. Odent. hp-clouds an h-p meshless method. *Numerical Methods for Partial Differential Equations*, 12:673–705, 1996.

[77] C.T. Dyka and R.P. Ingel. An approach for tension instability in smoothed particle hydrodynamics (sph). *Computers and Structures*, 57(4):573 – 580, 1995.

[78] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[79] ESI. https://www.esi-group.com/software-solutions/virtual-environment/cfd-multiphysics.

[80] R. Eymard, T. Gallouet, and R. Herbin. *Handbook of Numerical Analysis*, volume 7, chapter Finite volume method, pages 713–1020. P.G. Ciarlet and J.L. Lions, 2000.

[81] G. Fairweather and A. Karageorghis. The method of fundamental solutions for elliptic boundary value problems. *Advances in Computational Mathematics*, 9(1-2):69–95, 1998.

[82] H. Feng, R. Van der Wijngaart, and R. Biswas. Unstructured adaptive meshes: bad for your memory? *Appl. Numer. Math.*, 52(2-3):153–173, February 2005.

[83] B.A Finlayson. *The method of weighted residuals and variational principles: With application in fluid mechanics, heat and mass transfer*. Academic Press (New York), 1972.

[84] World footwear. Throw away the clay as modelling goes digital. *Magazine world footwear*, April 2003.

[85] P.J. Frey and P-L. George. *Mesh generation: application to finite elements*. 1st edition. Hermes science, 2000.

[86] T-P. Fries and T. Belytschko. Convergence and stabilization of stress-point integration in mesh-free and particle methods. *International Journal for Numerical Methods in Engineering*, 74(7):1067–1087, 2008.

[87] I. Gaponov, Jee-Hwan Ryu, Seong-Joo Choi, Hyun-Chan Cho, and Yury Poduraev. Telerobotic system for cell manipulation. In *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 165–169, July 2008.

[88] Christoph Gauger, Peter Leinen, and Harry Yserentant. The finite mass method. *SIAM J. NUMER. ANAL*, 37:1768–1799, 1999.

[89] M. Gauthier and S. Régnier. *Robotic Micro-Assembly*. Wiley-IEEE Press.

[90] F.C. Günther and W.K. Liu. Implementation of boundary conditions for meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 163(1-4):205 – 230, 1998.

[91] C. Gignoux and B. Silvestre-Brac. Hamilton's principle. In *Solved Problems in Lagrangian and Hamiltonian Mechanics*, pages 111–164. Springer Netherlands, 2009.

[92] R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Mon. Not. Roy. Astron. Soc.*, 181, November 1977.

[93] R.A Gingold and J.J Monaghan. Kernel estimates as a basis for general particle methods in hydrodynamics. *Journal of Computational Physics*, 46(3):429 – 453, 1982.

[94] Numerical Modelling GmbH. http://nmtec.ch/nm-seses/.

[95] J.I. Craig G.M.L. Gladwell, O.A. Bauchau. Variational and energy principles. In *Structural Analysis*, volume 163 of *Solid Mechanics and its Applications*, pages 673–717. Springer Netherlands, 2009.

[96] J.E. Goodman and J. J. O'Rourke. *Handbook of Discrete and Computational Geometry, Second Edition*. Discrete Mathematics and Its Applications Series. Chapman & Hall, 2004.

[97] J. Gosz. Admissible approximations for essential boundary conditions in the reproducing kernel particle method. *Computational Mechanics*, 19:120–135, 1996.

[98] S. Gottschalk, M.C. Lin, and D. Manocha. Obbtree: A hierarchical structure for rapid interference detection. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 171–180, New York, NY, USA, 1996. ACM.

[99] Silicon Graphics. The OpenGL Graphics System: a specification,noteVersion 2.1 2006.

[100] A. Gregory, M. Lin, S. Gottschalk, and R. Taylor. H-collide: A framework for fast and accurate collision detection for haptic interaction. Technical report, Chapel Hill, NC, USA, 1998.

[101] A. Gregory, A. Mascarenhas, S. Ehmann, M. Lin, and D. Manocha. Six degree-of-freedom haptic display of polygonal models. In *Proceedings of the Conference on Visualization '00*, VIS '00, pages 139–146, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press.

[102] M. Griebel and M.A Schweitzer. A particle-partition of unity method–part ii: Efficient cover construction and reliable integration. *SIAM J. Sci. Comput.*, 23:1655–1682, May 2001.

[103] M. Griebel and M.A. Schweitzer. A particle-partition of unity method—Part V: Boundary conditions. In S. Hildebrandt and H. Karcher, editors, *Geometric Analysis and Nonlinear Partial Differential Equations*, pages 517–540. Springer, 2002.

[104] M. Guiatni, V. Riboulet, C. Duriez, A. Kheddar, and S. Cotin. A Combined Force and Thermal Feedback Interface for Minimally Invasive Procedures Simulation. *IEEE/ASME Transactions on Mechatronics*, 18(3):1170–1181, June 2013.

[105] S. Hagihara and S. Motoda. Baisc discussion of boundary conidtion of smootherd particule hydrodynamics for analysis of cerebral contusion. In G.R. Liu, V.B.C. Tan, and X. HAN, editors, *Computational Methods*, pages 1435–1439. Springer Netherlands, 2006.

[106] G.A Hansen, R.W Douglass, and A Zardecki. *Mesh enhancement: Selected Elliptic Methods, Foundations and Applications*. 1st edition. Imperial College Press, March 2005.

[107] R.L. Hardy. Multiquadric equations of topography and other irregular surfaces. *Jounarl of geophycis research*, 176:1905–1915, 1971.

[108] R.L. Hardy. Theory and applications of the multiquadric-biharmonic method, 20 years of discovery 1968-1988. *Computers & Mathematics with Applications*, 19(8-9):163 – 208, 1990.

[109] F. Hartmann. The discrete babuška-brezzi condition. *Ingenieur-Archiv*, 56(3):221–228, 1986.

[110] V. Hayward and O.R. Astley. *Robotics Research: The Seventh International Symposium*, chapter Performance Measures for Haptic Interfaces, pages 195–206. Springer London, London, 1996.

[111] V. Hayward, O.R. Astley, M.Gruz-Hernandez, Danny Grant, and G. Robles de la Torre. Haptic interfaces and devices. *Sensor review,Emerald Group Publishing Limited*, 24(1):16–29, 2004. ISSN 0260-2288,.

[112] T.K Hellen. *Advanced non-linear finite element Analysis.NAFEMS course 2010*. NAFEMS, 2010.

[113] D. L. Hicks, J. W. Swegle, and S. W. Attaway. Conservative smoothing stabilizes discrete-numerical instabilities in sph material dynamics computations. *Applied Mathematics and Computation*, 85(2-3):209 – 226, 1997.

[114] C.H. Ho, C. Basdogan, and M.A. Srinivasan. Efficient point-based rendering techniques for haptic display of virtual objects. *Presence: Teleoper. Virtual Environ.*, 8(5):477–491, October 1999.

[115] C.H. Ho, C. Basdogan, and M.A. Srinivasan. Ray-based haptic rendering: Force and torque interactions between a line probe and 3d objects in virtual environments. *The International Journal of Robotics Research*, 19(7):668–683, 2000.

[116] C.H. Ho, C. Basdogan, M.A. Srinivasan, and A.A. Srinivasan. Haptic rendering: Point and ray-based interactions. In *Proceedings of the Second PHANTOM Users Group Workshop*, pages 13–18, 1997.

[117] R.W. Hockney and J.W. Eastwood. *Computer simulation using particles*. Taylor and Francis, Inc., Bristol, PA, USA, 1988.

[118] W. Holger. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995.

[119] R. L. Hollis, S. Salcudean, and D. W. Abraham. Toward a tele-nanorobotic manipulation system with atomic scale force feedback and motion resolution. In *Micro Electro Mechanical Systems, 1990. Proceedings, An Investigation of Micro Structures, Sensors, Actuators, Machines and Robots. IEEE*, pages 115–119, Feb 1990.

[120] W.M. Hsu, J.F Hughes, and H. Kaufman. Direct manipulation of free-form deformations. *SIGGRAPH Comput. Graph.*, 26(2):177–184, July 1992.

[121] P. Hubbard. Interactive collision detection. In *In Proceedings of IEEE Symposium on Research Frontiers in Virtual Reality*, pages 24–31, 1993.

[122] K.H Huebner, D.L Dewhirst, D.E. Smith, and T.G Byrom.

[123] A. Huerta and S. Fernández-Méndez. Enrichment and coupling of the finite element and meshless methods. *International Journal for Numerical Methods in Engineering*, 48(11):1615–1636, 2000.

[124] A. Huerta, S. Fernández-Méndez, and W.K. Liu. A comparison of two formulations to blend finite elements and mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):1105 – 1117, 2004. Meshfree Methods: Recent Advances and New Applications.

[125] T.J.R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover, New York, 2000.

[126] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135 – 4195, 2005.

[127] I. W. Hunter, S. Lafontaine, P. M. F. Nielsen, P. J. Hunter, and J. M. Hollerbach. Manipulation and dynamic mechanical testing of microscopic objects using a tele-micro-robot system. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, volume 3, pages 1553–1558, May 1989.

[128] J.M. Hyman, R.J. Knapp, and J.C. Scovel. High order finite volume approximations of differential operators on nonuniform grids. *Physica D: Nonlinear Phenomena*, 60(1-4):112 – 138, 1992.

[129] F. Jaillet, B. Shariat, and D. Vandorpe. Deformable object reconstruction with particle systems. *Computers & Graphics*, 22(2-3):189 – 194, 1998.

[130] D.L. James and D.K. Pai. ArtDefo - Accurate Real Time Deformable Objects. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 65–72, Los Angeles, 1999. Addison Wesley Longman.

[131] R.S. Johansson and R.H. Lamotte. Tactile detection thresholds for a single asperity on an otherwise smooth surface. *Somatosensory Research*, 1(1):21–31, 1983.

[132] E.J. Kansa. Multiquadrics-a scattered data approximation scheme with application to computational fluid dynamics. .*Computer Math. Appl*, 19:147–161, 1990.

[133] E.J. Kansa. Motivation for using radial basis functions to solve pdes. Technical report, Lawrence Livermore national laboratory and Embry-riddle Aeronatical University, 1999.

[134] M.J. Kilgard. The OpenGL utility toolkit (GLUT),programming interface API version3. Silicon Graphics,1996.

[135] Laehyun Kim. *An Implicit-based Haptic Rendering Technique*. PhD thesis, Los Angeles, CA, USA, 2003.

[136] T. Kim and M.R. Flynn. Numerical simulation of air flow around multiple objects using the discrete vortex method. *Journal of Wind Engineering and industrial Aerodynamics*, 56:213–234, August 1994.

[137] G. Kiss, C. Giannelli, U. Zore, B. Juttler, D. Grobmann, and J. Barner. Adaptive {CAD} model (re-)construction with thb-splines. *Graphical Models*, 76(5):273 – 288, 2014. Geometric Modeling and Processing 2014.

[138] R. L. Klatzky, D. Pawluk, and A. Peer. Haptic perception of material properties and implications for applications. *Proceedings of the IEEE*, 101(9):2081–2092, Sept 2013.

[139] Roberta Klatzky. Haptic perception and implications for design. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[140] J.G. Korvink and O. Paul. *MEMS, a practical guide to design, analysis and applications*. William Andrew, 2006.

[141] Y. Krongauz and T. Belytschko. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering*, 131(1-2):133 – 145, 1996.

[142] A. Kunoth. Multilevel preconditioning Ů appending boundary conditions by lagrange multipliers. *Advances in Computational Mathematics*, 4(1):145–170, 1995.

[143] V.D. Kupradze and M.A. Aleksidze. The method of functional equations for the approximate solution of certain boundary value problems. *{USSR} Computational Mathematics and Mathematical Physics*, 4(4):82 – 126, 1964.

[144] K. Laehyun, A. Kyrikou, and M.Desbrun G.S.Sukhatme. An implicit-based haptic rendering technique. Technical report, University of Southern California, 2002.

[145] L. Lan, J. Feng, F. Jian-ren, and C.ke-Fa. Recent developement of vortex method in incompressible viscous bluff body flows. *Journal of Zhejiang university science*, 6(4):283–288, 2005.

[146] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods p lancaster, k salkauskas. *Mathematics of Computation*, 37(155):141–158, 1981. Publisher: American Mathematical Society.

[147] H.P. Langtangen. *Computational Partial Differential Equations Numerical Methods and diffpack Programming*, volume 2. Springer, 1999.

[148] P.D. Lax and A.N. Milgram. Parabolic equations in contributions to the theory of partial differential equations. *Annals of Mathematics Studies*, 33:167–190, 1964. Princeton University Press.

[149] D.T. Lee and F.P. Preparata. Computational geometry – a survey. *IEEE TRANSACTIONS ON COMPUTERS*, 33(12):1072–1101, 1984.

[150] Y.T. Lee, A. Pennington, and N.K. Shaw. Automatic finite-element mesh generation from geometric models. a point-based approach. *ACM Trans. Graph.*, 3(4):287–311, oct 1984.

[151] P. Leinen. Realization of the finite mass method. In Peter Sloot, Alfons Hoekstra, C. Tan, and Jack Dongarra, editors, *Computational Science - ICCS 2002*, volume 2330 of *Lecture Notes in Computer Science*, pages 470–479. Springer Berlin / Heidelberg, 2002.

[152] A. Leonard. Review:vortex methods for flow. *Journal of Computational Physic*, 37(3):289–335, 1980.

[153] L. Libersky, D. Larry, A.G. Petschek, T.C. Carney, J.R. Hipp, and F.A. Allahdadi. High strain lagrangian hydrodynamics: a three-dimensional SPH code for dynamic material response. *J. Comput. Phys.*, 109(1):67–75, 1993.

[154] L. Libersky and A. Petschek. *Smooth particle hydrodynamics with strength of materials*, volume 395 of *Lecture Notes in Physics*, pages 248–257. Springer Berlin / Heidelberg, 1991.

[155] M.C. Lin. Efficient collision detection for animation and robotics, phd thesis. Technical report, University of California, Berkley,CA, 1993.

[156] T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, 11(1-2):83 – 95, 1980.

[157] T.J. Liszka. Hp-meshless cloud method for dynamic fracture in fluid interaction. Technical report, Altair Engineering, 2000.

[158] T.J. Liszka, C.A.M Duarte, and W.W. Tworzydlo. hp-meshless could method. *Computational methods for applied mechanics Engineering*, 139:263–288, 1996.

[159] G.R. Liu. *Moving beyond the Finite Element method*. CRC Press, USA, crc edition, 2002.

[160] G.R Liu and Y.T. Gu. A meshfree method: meshfree weak-strong (MWS) form method, for 2-d solids. *Computational Mechanics*, 33(1):2–14, 2003.

[161] G.R Liu and M.B Liu. Smoothed particle hydrodynamics, a meshfree particle method. *World scientific*, 2003. Singapore.

[162] G.R. Liu and Y.T Liu. *An introduction to Meshfree methods and their programming*. Springer, 2005.

[163] G.R. Liu, Y.L. Wu, and H. Ding. Meshfree weak-strong (MWS) form method and its application to incompressible flow problems. *International Journal for Numerical Methods in Fluids*, 46(10):1025–1047, 2004.

[164] G.R. Liu, G.Y. Zhang, Y.Y. Wang, Z.H. Zhong, G.Y. Li, and X. Han. A nodal integration technique for meshfree radial point interpolation method (ni-rpim). *International Journal of Solids and Structures*, 44(11-12):3840 – 3860, 2007.

[165] M.B. Liu and G.R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Journal Archives of Computational Methods in Engineering*, 17(1):25–76, March 2010.

[166] M.B. Liu, G.R. Liu, and K.Y. Lam. Constructing smoothing functions in smoothed particle hydrodynamics with applications. *Journal of Computational and Applied Mathematics*, 155(2):263 – 284, 2003.

[167] W.K. Liu, J. Adee, and S. Jun. Reproducing kernel particle methods for elastic and plastic problems. *Advanced Computational Methods for Material Modeling, Eds. Benson, D. J. , and Asaro*, pages 175–190, 1993.

[168] W.K. Liu and Y. Chen. Wavelet and multiple scale reproducing kernel methods. *International Journal for Numerical Methods in Fluids*, 21(10):901–931, 1995.

[169] W.K. Liu, S. Jun, S. Li, J. Adee, and T. Belyschko. Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering*, 38(1):1655–1679, 1994.

[170] W.K. Liu, S. Jun, and Y.F. Zhang. Reproducing kernel particle methods. *International Journal of Numerical Methods in Fluids*, 20:1081–1106, 1995.

[171] X. Liu, G.R. Liu, K.Tai, and K.Y Lam. radial point interpolation collocation method for the solution of partial differential equations. *Computer and mathematics with apllication*, 50:1425–1442, 2005.

[172] X. Liu and K. Tai. Polynomial point interpolation collocation method for the solution of partial differential equations. *Engineering analysis with boundary elements*, 30:598–609, 2006.

[173] R. Lohner, C.Sacco, E.Onate, and Sergio Idelsohn. A finite point method for compressible flow. *Computational mechanics*, 21:283–292, 1998.

[174] L.B. Lucy. A numerical approach to testing the fission hypothesis. *The astronomical journal*, 82(12):1013–1024, December 1977.

[175] Q. Luo and J. Xiao. Contact and deformation modeling for interactive environments for interactive environments. *IEEE TRANSACTIONS ON ROBOTICS*, 23(3), june 2007.

[176] E. Madenci and I. Guven. Finite element equations. In *The Finite Element Method and Applications in Engineering Using Ansys*, pages 187–296. Springer US, 2006.

[177] W.R. Madych. Miscellaneous error bounds for multiquadric and related interpolators. *Computers & Mathematics with Applications*, 24(12):121 – 138, 1992.

[178] W.R. Madych and S.A. Nelson. Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation. *Journal of Approximation Theory*, 70(1):94 – 114, 1992.

[179] A. Malek and S. Hodjatollah Momeni-Masuleh. A mixed collocation finite difference method for 3d microscopic heat transport problems. *Journal of Computational and Applied Mathematics*, 217(1):137 – 147, 2008.

[180] Nguyen Dang Manh, Anton Evgrafov, Allan Roulund Gersborg, and Jens Gravesen. Isogeometric shape optimization of vibrating membranes. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1343 – 1353, 2011.

[181] W. A. McNeely, K.D. Puterbaugh, and J.J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 401–408, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[182] U. Meier, O. López, C. Monserrat, M.C. Juan, and M. Alcañiz. Real-time deformable models for surgery simulation: a survey. *Computer Methods and Programs in Biomedicine*, 77(3):183 – 197, 2005.

[183] J.M. Melenk. *On Generalized Finite Element Methods*. Phd thesis, University of Maryland, 1995.

[184] J.M. Melenk and I. Babuska. The partition of unity finite element method. basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 196:289–314, 1996.

[185] G. Millet, S. Haliyo, S. Regnier, and V. Hayward. The ultimate haptic device: First step. In *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, pages 273–278, March 2009.

[186] B. Mirtich. V-clip: Fast and robust polyhedral collision detection. *ACM Trans. Graph.*, 17(3):177–208, July 1998.

[187] J.J. Monaghan. Why particle works? *SIAM J. Sci. and Stat. Comput.*, Volume 3(Issue 4):422–433, 1982.

[188] J.J. Monaghan. On the problem of penetration in particle methods. *J. Comput. Phys.*, 82(1):1–15, 1989.

[189] J.J. Monaghan. Modelling the universe. *Astronomical Society of Australia, Proceedings (ISSN 0066-9997)*, 8(3):233–237, 1990.

[190] J.J. Monaghan. An introduction to sph. *Computer Physics Communications*, 48(1):89–96, January 1998.

[191] J.J. Monaghan. Sph without a tensile instability. *J. Comput. Phys.*, 159(2):290–311, 2000.

[192] J.J. Monaghan and R.A. Gingold. Shock simulation by the particle method sph. *Journal of Computational Physics*, 52(2):374 – 389, 1983.

[193] A. B. Mor and T. Kanade. Modifying soft tissue models: Progressive cutting with minimal new element creation. In *In MICCAI*, pages 598–607. Springer-Verlag, 2000.

[194] K. Morton and D. Mayers. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, 1994.

[195] Muhanna A. Muhanna. Virtual reality and the cave: Taxonomy, interaction challenges and research directions. *Journal of King Saud University - Computer and Information Sciences*, 27(3):344 – 361, 2015.

[196] Y.X. Mukherjee and S. Mukherjee. The boundary node method for potential problems. *International Journal for Numerical Methods in Engineering*, 40(5):797–815, 1997.

[197] B. Muravin and E. Turkel. Multiple crack weight for solution of multiple interacting cracks by meshless numerical methods. *International Journal for Numerical Methods in Engineering*, 67(8):1146–1159, 2006.

[198] Attila P. Nagy, Mostafa M. Abdalla, and Zafer Gürdal. Isogeometric sizing and shape optimisation of beam structures. *Computer Methods in Applied Mechanics and Engineering*, 199(17-20):1216 – 1230, 2010.

[199] C.S. Nam, L.W. Borsodi, W.A. Mackay, and J. Di. A haptic thermal interface: Towards effective multimodal user interface systems. In *In Proceedings of IASTED-HCI, ACTA Press*, pages 13–18, 2005.

[200] D. Nardini and C.A. Brebbia. A new approach to free vibration analysis using boundary elements. *Applied Mathematical Modelling*, 7(3):157 – 162, 1983.

[201] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Journal of Computational Mechanics*, 10:307–318, 1992.

[202] Vinh Phu Nguyen, Cosmin Anitescu, Stéphane P.A. Bordas, and Timon Rabczuk. Isogeometric analysis: An overview and computer implementation aspects. *Mathematics and Computers in Simulation*, 117:89 – 116, 2015.

[203] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics, 1992.

[204] Han-Wen Nienhuys and Frank A. van der Stappen. Supporting Cuts and Finite Element Deformation in Interactive Surgery Simulation. *Proceedings Medical Image Computing and Coputer Assisted Intervention*, 2001.

[205] P. Novo. *Auditory Virtual Environments*, pages 277–297. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[206] J.T. Oden and G.F. Carey. *Finite Element:Mathematical aspects Volume IV*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

[207] E. Onate and S idelsohn. A mesh-free finite point method for advective-diffusive transport and fluid flow problems. *Computational mechanics*, 21:283–292, 1998.

[208] E. Onate, S Idelsohn, O.C.Zienkiewicz, and R.L.Taylor. A finite point method in computational mechanics.applcations to convective transport and fluid flow. *International journal for numerical methods in Engineeering*, 39(22):3839–3866, December 1998.

[209] E. Onate and F Perazzo. A finite point method for elasticity problems. *Computer and structures*, 79:2151–2163, 2001.

[210] E. Onate, S.Idelsohn, O.C.Zienkiewicz, R.L.Taylor, and C.Sacco. A stabilized finite point method for analysis of fluid mechanics problems. *Computer methods in applied mechanics and engineering*, 139:315–346, 1996.

[211] D. Organ, M. Fleming, T. Terry, and T. Belytschko. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational Mechanics*, 18(3):225–235, 1996.

[212] S. Owen. An introduction to mesh generation algorithms. In Byron W, editor, *Proceedings of the 14th International Meshing RoundtableHanks*, volume XIV. Springer, 2005.

[213] S.J. Owen. A survey of unstructured mesh generation technology. *Meshing Software Survey*, 1998. web page: http://www.andrew.cmu.edu/user/sowen/softsurv.html.

[214] P.W. Partridge and B. Sensale. The method of fundamental solutions with dual reciprocity for diffusion and diffusion-convection using subdomains. *Engineering Analysis with Boundary Elements*, 24(9):633 – 641, 2000.

[215] Christoph J. Paulus, Lionel Untereiner, Hadrien Courtecuisse, Stéphane Cotin, and David Cazier. Virtual cutting of deformable objects based on efficient topological operations. *The Visual Computer*, 31(6):831–841, 2015.

[216] Shahram Payandeh, Hilary Zhang, and Jekeon Cha. Towards interactive haptic simulation of cutting. *Int. J. Virtual Technol. Multimedia*, 1(2):172–186, March 2010.

[217] P.Berczik and S.G. Kravchuk. 3d modelling of forming galaxies; the history of star formation activity. *Astrophysics and Space Science*, Volume 245(Issue 1):pp.27–42, 1996.

[218] Guillaume Picinbono, Jean-Christophe Lombardo, Hervé Delingette, and Nicholas Ayache. Improving Realism of a Surgery Simulator : Linear Anisotropic Elasticity, Complex Interactions and Force Extrapolation. Research Report RR-4018, INRIA, 2000.

[219] J. Pitkäranta. Boundary subspaces for the finite element method with lagrange multipliers. *Numerische Mathematik*, 33:273–290, 1979.

[220] J. Pitkäranta. The finite element method with lagrange multipliers for domains with corners. *Mathematics of Computation*, 37(155):13–30, 1991.

[221] Plantsee. https://www.plansee.com/en/materials/tantalum.html.

[222] Seppo Pohja and Neuro Sics. Survey of studies on tactile senses. Technical report, March 1996.

[223] J.P. Ponthot and T. Belytschko. A coombined Element-Free Galerkin Method/ arbitrary Lagrangian-Eulerian formulation for dynamic crack propagation. In D. Durban and J.R.A. Pearson, editors, *IUTAM Symposium on Non-linear Singularities in Deformation and Flow*, pages 205–216. Springer Netherlands, 1999.

[224] P.Rademacher. GLUI: a Glut-based user interface library.

[225] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B. P. Flannery. *Numerical recipes*. Cambridge university press, 3rd edition, 2007.

[226] M. Puso, E. Zywicz, and J.S. Chen. A new stabilized nodal integration approach. In T.J. Barth, M. Griebel, D.E. Keyes, R.M. Nieminen, D. Roose, T.Schlick, and M.A. Schweitzer,

editors, *Meshfree Methods for Partial Differential Equations III*, volume 57 of *Lecture Notes in Computational Science and Engineering*, pages 207–217. Springer Berlin Heidelberg, 2007.

[227] P.W.Randles, L.D.Libersky, and A.G.Petchek. On neighbors, derivatives and viscosity in particles codes. In Los alamos national laboratory, editor, *Proceeding ECCM*, August 31-Spetember 3 1999. Munich Germany.

[228] T. Rabczuk, T. Belytschko, and S.P. Xiao. Stable particle methods based on lagrangian kernels. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):1035 – 1063, 2004. Meshfree Methods: Recent Advances and New Applications.

[229] P.W. Randles and L.D. Libersky. Smoothed particle hydrodynamics: Some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):375 – 408, 1996.

[230] P.W. Randles and L.D. Libersky. Normalized SPH with stress points. *International Journal for Numerical Methods in Engineering*, 48(10):1097–0207, 2000.

[231] J.N. Reddy. *An introduction to The Finite Element Method*. McGraw-Hill, New York, 1993.

[232] A.G. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Comput. Surv.*, 12(4):437–464, December 1980.

[233] A.G. Requicha and H.B. Voelcker. *Constructive Solid Geometry*. Technical memorandum, Production Automation Project, College of Engineering and Applied Science, University of Rochester. PAP, 1982.

[234] L.F. Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 210:307–357, 1911.

[235] R.D. Richtmyer and K.W. Morton. *Difference Methods for Initial Value Problems*. Wiley, New York, 2nd ed edition, 1967.

[236] C.M.C. Roque and A.J.M. Ferreira. Numerical experiments on optimal shape parameters for radial basis functions. *Numerical Methods for Partial Differential Equations*, 26(3):675–689, 2010. Singapore.

[237] Rosswog. Astrophysical smooth particle hydrodynamics. *New Astronomy Reviews*, 53(4-6):78 – 104, 2009.

[238] D.C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 345–352, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[239] J.K. Salisbury and M.A. Srinivasan. Sections on haptics. virtual environment technology for training. Technical report, The Virtual Environment and Teleoperator Research Consortium (VETREC) affiliated with MIT, 1992. BBN Report N. 7661.

[240] K. Salisbury, D. brook, N. Swarup, and C. Zilles. Haptic rendering: Programming touch interaction with virtual world. Technical report.

[241] Kenneth Salisbury and Christopher Tarr. Haptic rendering of surfaces defined by implicit functions. In *PROCEEDINGS OF THE ASME, DSC-61*, pages 61–67, 1997.

[242] H. Samet. *Spatial data structures: Quadtree, Octree and other hierarchical methods*. Addison Wesley, 1989.

[243] M.M. Sanders and E.J. McCormick. *Human factors in Engineering & design 7th ed.* McGraww-Hill, 1993.

[244] Sarah, Gibson, and Mirtich. A survey of deformable modeling in computer graphics. Technical report, MITSUBISHI ELECTRIC RESEARCH LABORATORIES, 1997.

[245] S.A. Sarra and E.J. Kansa. Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations. Technical report, Marshall University, 2009.

[246] R. Schaback. Comparison of radial basis function interpolants. In *Multivariate Approximation and Wavelets. World Scientific*, pages 1–12, Singapore, 1992.

[247] R. Schaback. Improved error bounds for scattered data interpolation by radial basis functions. *Math. Comp*, 68:201–216, 1999.

[248] R. Schaback and H. Wendland. Kernel techniques: From machine learning to meshless methods. *Acta Numerica*, 15(1):543–639, 2006.

[249] M. Schäfer. Finite-element methods. In *Computational Engineering, Introduction to Numerical Methods*, pages 107–148. Springer Berlin Heidelberg, 2006.

[250] Jean-Marc Schwartz, Marc Denninger, Denis Rancourt, Christian Moisan, and Denis Laurendeau. Modelling liver tissue properties using a non-linear visco-elastic model for surgery simulation. *Medical Image Analysis*, 9(2):103 – 112, 2005. Medical Simulation - DelingetteMedical Simulation - Delingette.

[251] M.A. Scott, R.N. Simpson, J.A. Evans, S. Lipton, S.P.A. Bordas, T.J.R. Hughes, and T.W. Sederberg. Isogeometric boundary element analysis using unstructured t-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197 – 221, 2013.

[252] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4):151–160, August 1986.

[253] Sensable. http://www.geomagic.com/en/products/phantom-omni/overview.

[254] Sensable. OPENHAPTIC toolkit. API reference.

[255] Sensable. OPENHAPTIC toolkit. programmer's guide.

[256] S.D. Senturia. CAD challenges for microsensors, microactuators, and microsystems. In *Proceedings of the IEEE*, pages 1611–1626, 1998.

[257] Y.Y. Shan, C.Shu, and N.Qin. Multiquadric finite difference method and its application. *Advances in applied mathematics and mechanics*, 1(5):615–638, 2009.

[258] Amit Shaw and D. Roy. Nurbs-based parametric mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 197(17-18):1541 – 1567, 2008.

[259] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *ACM '68: Proceedings of the 1968 23rd ACM national conference*, pages 517–524, New York, NY, USA, 1968. ACM.

[260] W. Sheppard. Central-difference formulae. *Proceeding of the London Mathematical society*, 31:449–488, 1899.

[261] J.R. Shewchuk. Triangle: Engineering a 2d quality mesh generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.

[262] R.N. Simpson, S.P.A. Bordas, J. Trevelyan, and T. Rabczuk. A two-dimensional isogeometric boundary element method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering*, 209-212:87 – 100, 2012.

[263] C.M. Smith. Human factors in haptic interfaces. *Crossroads*, 3(3):14–16, April 1997.

[264] G. Smith. *Numerical solution of partial differential equations*. oxFord University press, 1985-2004.

[265] I.M Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86 – 112, 1967.

[266] SoftMEMS. http://www.softmems.com/.

[267] M.A. Srinivasan. What is haptic? Technical report. Laboratory for human and Machine-Haptics: the Touch Lab.

[268] K.M. Stanney, R.R. Mourant, and R.S. Kennedy. Human factors issues in virtual environments: A review of the literature. *Presence: Teleoper. Virtual Environ.*, 7(4):327–351, August 1998.

[269] O. Steinbach. Variational formulations of boundary value problems. In *Numerical Approximation Methods for Elliptic Boundary Value Problems*, pages 59–87. Springer New York, 2008.

[270] R.F. Stellingwerf and C.A. Wingate. Impact modeling with smooth particle hydrodynamics. *International Journal of Impact Engineering*, 14(1-4):707 – 718, 1993.

[271] M.J. Stock. Summary of vortex methods litterature. Technical report, 2002-2007.

[272] H. Sundar, R.S Sampath, and G. Biros. Bottom-up construction and 2:1 balance refinement of linear octrees in parallel. *SIAM J. Sci. Comput.*, 30(5):2675–2708, August 2008.

[273] J.W. Swegle, S.W. Attaway, M.W. Heinstein, F.J. Mello, and D.L. Hicks. An analysis of smoothed particle hydrodynamics. *NASA STI/Recon Technical Report N*, 95:17439–+, March 1994. Provided by the SAO/NASA Astrophysics Data System.

[274] J.W. Swegle, D.L. Hicks, and S.W. Attaway. Smoothed particle hydrodynamics stability analysis. *J. Comput. Phys.*, 116(1):123–134, 1995.

[275] K.Y. Sze, J.S. Chen, N. Sheng, and X.H. Liu. Stabilized conforming nodal integration: exactness and variational justification. *Finite Elements in Analysis and Design*, 41(2):147 – 171, 2004.

[276] H. Talbot, F. Spadoni, C. Duriez, M. Sermesant, M. O 'neill, P. Jaïs, S. Cotin, and H. Delingette. Interactive Training System for Interventional Electrocardiology Procedures. *Medical Image Analysis*, 2016.

[277] H. Tan, J. Radcliffe, B.N. Ga, H.Z. Tan, B. Eberman, M.A. Srinivasan, and B. Cheng. Human factors for the design of force-reflecting haptic interfaces, 1994.

[278] J.C. Tannehill, D.A. Anderson, and R.H. Pletcher. *Computational fluid mechanics and Heat transfer*. Taylor and Francis, Washington DC, 2nd edition, 1997.

[279] B. Thomas T.B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, Cambridge, MA, USA, 1992.

[280] T.Belytschko, Y.Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[281] T.G. Terry. Fatigue crack propagation modeling using the element free Galerkin method, master's thesis. Technical report, Northwestern university, 1994.

[282] J. Thomas. *Numerical Partial Differential Equations: Finite Difference Methods*. Springer-Verlag, 1995.

[283] L. Thurfjell, J. Mclaughlin, J.Mattsson, and P.Lammertse. Haptic interaction with virtual objects: The technology and some applications. *Industrial Robot: An industrial journal*, 29(3):210–215, 2002.

[284] M.A.O. Tristan. 6-dof haptic rendering using contact levels of detail and haptic textures, phd thesis. Technical report, University of North Carolina, 2004.

[285] T. Tu, D.R. O'Hallaron, and O. Ghattas. Scalable parallel octree meshing for terascale applications. In *SC '05: Proceedings of the 2005 ACM/IEEE conference on Supercomputing*, page 4, Washington, DC, USA, 2005. IEEE Computer Society.

[286] R. van Liere, J.D. Mulder, and J.J. Van Wijk. Computational steering, 1997.

[287] R. Vignjevic and J. Campbell. Review of development of the smooth particle hydrodynamics (sph) method. *Predictive Modeling of Dynamic Processes*, pages 367–396, 2009.

[288] R. Vignjevic, J. Campbell, and L. Libersky. A treatment of zero-energy modes in the smoothed particle hydrodynamics method. *Computer Methods in Applied Mechanics and Engineering*, 184(1):67 – 85, 2000.

[289] G.J. Wagner and W.K. Liu. Application of essential boundary conditions in mesh-free methods: a corrected collocation method. *International Journal for Numerical Methods in Engineering*, 47(8):1367–1379, 2000.

[290] W.A. Wall, M.A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33-40):2976 – 2988, 2008.

[291] H. Wang, W. Dai, and L.G. Hewavitharana. A finite difference method for studying thermal deformation in a double-layered thin film with imperfect interfacial contact exposed to ultrashort pulsed lasers. *International Journal of Thermal Sciences*, 47(1):7 – 24, 2008.

[292] H. Wang, W. Dai, R. Nassar, and R. Melnik. A finite difference method for studying thermal deformation in a thin film exposed to ultrashort-pulsed lasers. *International Journal of Heat and Mass Transfer*, 49(15-16):2712 – 2723, 2006.

[293] J.G. Wang and G.R. Liu. On the optimal shape parameters of radial basis functions used for 2-d meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 191(23-24):2611 – 2630, 2002.

[294] H. Wendland. Error estimates for interpolation by compactly supported radial basis functions of minimal degree. *Journal of Approximation Theory*, 93(2):258 – 272, 1998.

[295] H. Wendland. Meshless Galerkin methods using radial basis functions. *Mathematics of computation*, 68(228):1521–1531, March 1999.

[296] Wikipedia. List of cad,cae and cam companies. http://en.wikipedia.org/wiki/$List\_of\_CAD\_companies$.

[297] J.D. Wood and H. Wright. Steering via the image in local, distributed and collaborative settings. *Concurrency and Computation: Practice and Experience*, 20(3):265–276, 2008.

[298] Z. Wu. Compactly supported positive definite radial functions. *Advances in Computational Mathematics*, 4(1):283–292, 1995.

[299] Z-M Wu and R. Schaback. Local error estimates for radial basis function interpolation of scattered data. *IMA J. Numer. Anal*, 13:13–27, 1992.

[300] H. Yserentant. A new class of particle methods. *Numerische Mathematik*, 76:87–109, 1997.

[301] J. Zhang and J.J. Zhao. Iterative solution and finite difference approximations to 3d microscale heat transport equation. *Mathematics and Computers in Simulation*, 57(6):387 – 404, 2001.

[302] J. Zhang and J.J. Zhao. Unconditionally stable finite difference scheme and iterative solution of 2d microscale heat transport equation. *Journal of Computational Physics*, 170(1):261 – 275, 2001.

[303] Jian Zhang, Shahram Payandeh, and John Dill. Levels of detail in reducing cost of haptic rendering: a preliminary user study, 2003.

[304] Z. Zhang. Derivative superconvergent points in finite element solutions of harmonic functions: a theoretical justification. *Math. Comput.*, 71(240):1421–1430, October 2002.

[305] J.X. Zhou, J.B. Wen, H.Y. Zhang, and L. Zhang. A nodal integration and post-processing technique based on Voronoi diagram for Galerkin meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 192(35-36):3831 – 3843, 2003.

[306] T. Zhu and S.N. Atluri. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. 2007.

[307] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. McGraw-Hill, New York, 1989.

[308] C.B. Zilles and J.K. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of the International Conference on Intelligent Robots and Systems-Volume 3 - Volume 3*, IROS '95, pages 3146–, Washington, DC, USA, 1995. IEEE Computer Society.